A Survey of Blockchain Consensus Protocols

JIE XU, CONG WANG, and XIAOHUA JIA, City University of Hong Kong, China

Blockchain consensus protocols have been a focus of attention since the advent of Bitcoin. Although classic distributed consensus algorithms made significant contributions to the development of blockchain consensus protocols, there are still many issues to be resolved due to the complexity and diversity of the blockchain. In this survey, we summarize the state-of-the-art blockchain consensus protocols. We first introduce the theoretical basis, models, and challenges of blockchain consensus protocols. Then, we present the existing blockchain protocols in the categories of proof-based protocols, committee-based protocols, and other miscellaneous protocols. Finally, we analyze their performance and discuss future research directions by comparing existing protocols.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Security and privacy → Distributed systems security**; • **Networks → Network protocol design**.

Additional Key Words and Phrases: blockchain consensus, cryptocurrency, security, scalability, decentralization

1 INTRODUCTION

Blockchain is a distributed ledger composed of a series of blocks built on a peer-to-peer network. The blockchain consensus protocol enables participating nodes to append blocks to the blockchain in the same and unique order [25]. However, malicious participating nodes may engage in arbitrary malicious behavior, which can lead to the failure of consensus. Therefore, simple Crash Fault Tolerance (CFT) [79, 98, 99] no longer applies to blockchains. Byzantine Fault Tolerance (BFT) [26, 80] is required in blockchain systems. Despite significant progress in the research of distributed consensus algorithms, the design of a secure and efficient BFT blockchain consensus protocol remains a critical and challenging task.

In 2008, Nakamoto proposed Bitcoin [96], a peer-to-peer electronic payment system that eliminates the need for trusted third parties. Bitcoin is the first implementation of the BFT consensus protocol in a permissionless environment. Its consensus protocol is called Nakamoto Consensus protocol. Nakamoto consensus protocol elects the leader through a Proof of Work (PoW) algorithm. Participating nodes compete to solve a puzzle, and the node that solves the puzzle first becomes the leader. The leader then has the right to generate a new block and append it to the blockchain. The longest chain is considered the authoritative blockchain when a blockchain forks [96].

Since Bitcoin, numerous consensus protocols have been proposed to address various issues in the blockchain. Some protocols [10, 49, 83, 129] aim to increase blockchain throughput and reduce transaction confirmation latency. Some protocols [57, 69, 71] aim to avoid the massive energy consumption of PoW. Some others [16, 92, 102, 117] focus on enhancing blockchain security.

There are two main design ideas for the blockchain consensus protocol. The first is proof-based consensus protocols, which mimic the design idea of Nakamoto consensus protocol. They are generally deployed in a permissionless environment, where enrollment is open to all nodes, and randomly assign the right of new block

Authors' address: Jie Xu, jiexu49-c@my.cityu.edu.hk; Cong Wang, congwang@cityu.edu.hk; Xiaohua Jia, csjia@cityu.edu.hk, City University of Hong Kong, Tat Chee Avenue, Hong Kong, China.

0360-0300/2023/8-ART111 $15.00
https://doi.org/10.1145/3579845

generation to participating nodes. Proof-based consensus protocols include PoW-based protocols [49, 83, 129], Proof of Stake (PoS)-based protocols [24, 57, 68, 69], Proof of Storage (PoStorage)-based protocols [51, 91, 100], etc. PoW-based, PoS-based, and PoStorage-based protocols give the participants with high computing power, stakes, and storage, respectively, a higher priority in winning the leader competition. Another idea is committee-based consensus protocols, following the classic consensus algorithms in distributed systems, such as PBFT [26]. Committee-based consensus protocols [26, 42, 93] allow participants to reach consensus through voting. Once the number of votes in favor of a new block exceeds a certain proportion of the committee, the new block is appended to the blockchain. Committee-based protocols generally require the number of participants to be known in advance, which is easier to deploy in a permissioned environment where enrollment is controlled. Both types of consensus protocols have their advantages, and some blockchains have absorbed the ideas of the two and adopted a hybrid approach [57, 74, 87]. Moreover, there are other miscellaneous consensus protocols designed for specific applications, such as resource constraints [31, 105], non-anonymous proof-based protocols [21, 58], data redaction [6, 39], etc.

The past ten years have witnessed the growing trend of blockchain. Much literature has been published on blockchain consensus protocols. This survey aims to provide a comprehensive introduction to blockchain consensus protocols, with a particular focus on state-of-the-art developments. The main contributions of this survey can be summarized as follows. First, we classify consensus protocols and present a systematic overview. Then, we highlight the challenges faced by current blockchain consensus protocols and introduce existing efforts to address these challenges. Finally, we compare the characteristics of different consensus protocol types and discuss promising directions for future research. We believe that our work can help readers gain a deeper understanding of blockchain consensus and contribute to this growing field of research.

The remainder of this survey is structured as follows. We first introduce the preliminaries of blockchain consensus in Section 2, and then present the design principles and criteria of blockchain consensus protocols in Section 3. Section 4 to 7 presents the existing consensus protocols. We compare the existing blockchain consensus protocols and discuss future research directions in Section 8 and 9, respectively. Finally, we conclude the paper in Section 10.

2 PRELIMINARIES OF BLOCKCHAIN CONSENSUS

2.1 System Model of Blockchain

2.1.1 Data Structures. A blockchain includes the following data structures:

- Transactions: A transaction is a transfer operation in cryptocurrency. A transaction must specify the senders, receivers, the amount of the transaction, and the signature of the senders.
- Blocks: A block consists of two parts: the block header and the block body. The block body contains a set of transactions. The block header records the hash value of the previous block, version, random number, target difficulty, and the root of the Merkle tree, etc.
- Blockchain: Blockchain is a sequence of blocks. Each block contains the hash of the previous block as a pointer pointing to the previous block, which forms a chain. The new block is always appended to the blockchain.

2.1.2 Blockchain Types. Blockchains can be divided into two types based on the authorization requirements for participating nodes.

- Permissionless Blockchain: There is no entity/mechanism in permissionless blockchains to manage the identity of participating nodes [63]. Permissionless blockchains are completely open and decentralized. Anyone can participate or exit the permissionless blockchain at any time [93, 101]. Anyone can read the transactions on the blockchain or broadcast transactions to be appended to the blockchain. Permissionless blockchains always need incentive mechanisms, such as mining rewards, to keep the system in operation [63].

- Permissioned Blockchain: There is an entity or a mechanism that manages the identities and permissions of participating nodes in permissioned blockchains [63]. Only authorized nodes can append transactions to the blockchain or read the transactions on the blockchain according to permission policy [1, 4]. It is easier to govern than the permissionless blockchain. Malicious behavior can be penalized after being caught [63]. Incentives are not necessary for a permissioned blockchain system.

2.2 Network Models

The design of consensus protocols is closely related to the assumptions of the network models, as the security of the consensus protocols is based on the assumption of network models. There are generally three types of network models [44] as the following:

2.2.1 Synchronous Network. In a synchronous network, a message is guaranteed to be delivered within a fixed upper bound, and the upper bound is known. That is, the message sent at t will be received at the latest $t + d$, and d is a known constant. Therefore, a protocol based on the synchronous network settings can proceed in rounds. It can be guaranteed that all nodes have received the message before starting the next new round. The assumption of the synchronous network is simple but it is difficult to implement because the actual underlying network is unreliable [60].

2.2.2 Partially Synchronous Network. In a partially synchronous network, a message can be guaranteed to be delivered within a fixed upper bound of time, but the value of the bound is unknown. The message delivery time cannot be used as a known parameter in the design of a consensus protocol. Partially synchronous consensus protocols are more practical, but asynchronous attacks [93] can make partially synchronous consensus protocols no progress.

2.2.3 Asynchronous Network. There is no fixed upper bound for message delivery in asynchronous networks. The message is only guaranteed to arrive eventually, but the arrival time is unpredictable [93]. The asynchronous network is the most practical network model without timing assumptions. The security of asynchronous blockchain does not depend on the delivery time of messages.

3 OVERVIEW OF BLOCKCHAIN CONSENSUS PROTOCOLS

3.1 Proof-based Consensus Process

In a proof-based consensus protocol [36, 96, 126], nodes provide proof of leadership to append a new block to the blockchain. First, miners validate a set of transactions from received transactions, and generate a Merkle tree for the validated transactions. The validated transactions and their Merkle tree are packed into the new block. Second, a node is selected as the leader through a cryptographic random algorithm to generate a new block. The proof of being selected as the leader is broadcast to the entire network for validation together with the new block. Third, nodes validate the new block after receiving it. The valid new block is appended to the blockchain and confirmed after a few blocks.

3.2 Committee-based Consensus Process

In a committee-based consensus protocol [26, 93, 128], nodes vote to decide the next block to be appended to the blockchain. First, the proposer multicasts a preparation block request to other participants. Second, participants reply to the proposer with their status. If the proposer receives ready messages from a sufficient number of participants, it enters the pre-commit phase. Third, participants broadcast their votes to commit the proposed block. If the number of commit responses agreeing to the new block exceeds the threshold, it means that the new block will be appended to honest nodes' blockchains.

3.3 Requirements of Consensus Protocols

There are several requirements for the design of consensus protocols, including security, scalability, and decentralization.

3.3.1 Security. The security of the blockchain consensus protocol implies two important properties: *persistency* and *liveness* [55, 101]. Persistence states that if a transaction exists on an honest node's local ledger with more than k-block deep, where k is the security parameters, the transaction will permanently exist at the same position on any other honest node's ledger [55]. Persistence can ensure that the blockchains of all honest nodes keep consistency except for the last k blocks.

Persistence alone is not enough to guarantee the correct operation of a blockchain system [55], because adversaries can hinder consensus so that the blockchain stops growing. Liveness is another important property. It means that transactions generated by the honest nodes will eventually appear at a depth of more than k blocks on the blockchain of any honest node. Thus, blockchain is always growing towards a predetermined goal.

3.3.2 Scalability. Scalability refers to the efficiency of the system as the system scale and workload increase. There are two metrics to measure blockchain scalability: transaction throughput and transaction confirmation latency. Throughput refers to the rate at which transactions are appended to the blockchain. Transaction confirmation latency refers to the time from when the transaction is appended to the blockchain to when the transaction is considered to be irreversible with overwhelming probability.

3.3.3 Decentralization. Decentralization involves the distribution of resources, benefits, and so on. Decentralization can avoid corruption and collusion and help build a fairer system. The main metric of decentralization is the distribution of resources. For example, the distribution of computing power controlled by independent individuals or organizations, the distribution of stakes at different accounts, etc. The lower the proportion of the computing power controlled by organizations, or the fewer accounts with a large number of stakes, the better decentralization.

In the following sections, we first introduce proof-based consensus protocols for permissionless blockchains, then introduce committee-based consensus protocols for permissioned blockchains and permissionless blockchains, respectively. Finally, we discuss other miscellaneous protocols. Although we divide blockchain protocols into three categories, this classification is only one of many. For some hybrid consensus protocols, we will only focus on them in one of the categories they belong to and will not repeat.

4 POW-BASED BLOCKCHAIN CONSENSUS PROTOCOLS

Blockchains in a permissionless environment are vulnerable to Sybil attacks [101]. An adversary may pretend to be multiple nodes simultaneously to take advantage in the leader election. Nakamoto is the first permissionless blockchain to introduce PoW to resist Sybil attacks. After that, many PoW-based protocols were proposed to improve the performance of Nakamoto protocol from different aspects.

In this section, we first introduce the principles of Nakamoto Consensus Protocol. Then, we introduce existing work dedicated to improving the scalability, decentralization and security of Nakamoto consensus protocol, respectively. The comparison of PoW-based consensus protocols is summarized in Table 1.

Table 1. Comparison of PoW-based consensus protocols

Schemes	Aims	Methods	Comments
Bitcoin-NG [49]	Scalability	Decoupling functions	Vulnerable to DoS
Prism [10]	Scalability	Decoupling functions & parallel chains	Scalable to physical limits
OHIE [129]	Scalability	Parallel chains	Same theoretical security as Bitcoin
Chainweb [90]	Scalability	Parallel chains	Not resistant to sub-chain attacks
CliqueChain [70]	Scalability	Parallel chains	No throughput improvement
Monoxide [121]	Scalability	Multiple Chains & multiple zones	Chu-ko-nu amplifies mining power
Ethereum [123]	Scalability	DAG	Modified GHOST implementation
Spectre [113]	Scalability	DAG	Failed to serialize transaction globally
Phantom [85]	Scalability	DAG	Vulnerable to liveness attacks
Conflux [83]	Scalability	DAG	Resistant liveness attacks
Occam [125]	Scalability	DAG	Adaptively scale up and down
Miller *et al.* [92]	Decentralization	Nonoutsourceable & puzzles	Formally decentralization solution
Fruitchain [102]	Decentralization & security & fairness	Reward for fruits	Resistant selfish mining attacks
StrongChain [117]	Decentralization & security & fairness	Reward for weak puzzles	Resistant selfish mining attacks
Bobtail [16]	Security	Mean of the k-lowest order	Resistant double-spending attacks & selfish mining attacks

4.1 Nakamoto Consensus Protocol

Bitcoin nodes that participate in generating blocks are called miners. Only miners who solve puzzles can get the right to generate blocks. Huge computing power is required to solve the puzzles. As a result, it is difficult for adversaries to launch Sybil attacks by generating many Sybil nodes [53].

The puzzle in Bitcoin is to compute a *nonce* that meets the following conditions:

$$H(h_{i-1}, nonce, tx, par) < Target, \tag{1}$$

where h_{i-1} denotes the hash of the previous block, tx denotes the set of validated transactions, and par denotes other parameter information such as blockchain version and cryptographic parameters. $H()$ is a one-way hash function. $Target$ is a parameter of the puzzle difficulty, which is adjusted periodically according to actual block generation intervals and expected block generation intervals. The expected block generation intervals are around ten minutes. If actual block generation intervals are shorter, the difficulty increases; if actual block generation intervals are longer, the difficulty decreases. Therefore, regardless of the computing power in the network, Bitcoin can always keep around ten minutes to generate a new block.

Since it takes some time for a new block to reach other nodes, some miners may also successfully solve the puzzle before receiving the new block. This leads to the case that two new blocks extend from the same previous block. This case is called the temporary fork of the blockchain. To handle blockchain forks and maintain the single-chain structure, the Longest Chain Rule (LCR) is introduced. When multiple blocks are appended to the same block concurrently, miners should follow the longest chain. The longest blockchain is considered the authoritative one that concentrates more computing power. Miners keep extending the longest blockchain. Thus, although there might be forks at some point, over a longer time, one branch will become the longest. Because the longest chain is authoritative, the computing costs of modifying a block increase as the number of appended blocks increases. It is assumed that the possibility of modifying the block that has been appended by six blocks is negligible [96]. Therefore, transactions are considered to be confirmed after six blocks are appended in Bitcoin.

4.2 Protocols for Improving Scalability

Throughput and transaction confirmation latency of Bitcoin cannot meet users' demands, resulting in a large number of transactions congestion and a surge in transaction fees. The Bitcoin performance bottleneck is caused by many factors. At the consensus layer, in order to maintain the single-chain structure of the blockchain, all concurrent blocks must be serialized, which is the bottleneck of blockchain throughput. An intuitive idea to improve blockchain scalability is to allow blocks to be generated and broadcast in parallel. Based on this idea, many improved consensus protocols have been proposed. The specific implementation methods include decoupling blockchain functions [10, 49, 74], parallel-chain structure [121, 129], and DAG-based structure [83, 85], etc.

4.2.1 Decoupling Blockchain Functions. Bitcoin-NG [49] decouples the functions of blocks into *key blocks* for leader election and *microblocks* for transaction packing. A miner who successfully solves the puzzle becomes the leader of an epoch. The leader first generates a key block and starts a new epoch as shown in Figure 1. The key block does not contain any transactions but contains the solution to the puzzle and the leader's public key. The leader packs transactions to generate a sequence of microblocks without solving puzzles and appends them after the key block. The intervals between key blocks, i.e. an epoch are around ten minutes, but the intervals between microblocks within an epoch are around ten seconds. In this way, Bitcoin-NG improves the throughput but avoids forks caused by increasing the block size or decreasing the block interval. However, it does not reduce the transaction confirmation latency [10], and brings about problems such as the elected leader may be compromised [16], selfish mining attacks [16], or double-spending attacks [74], etc.

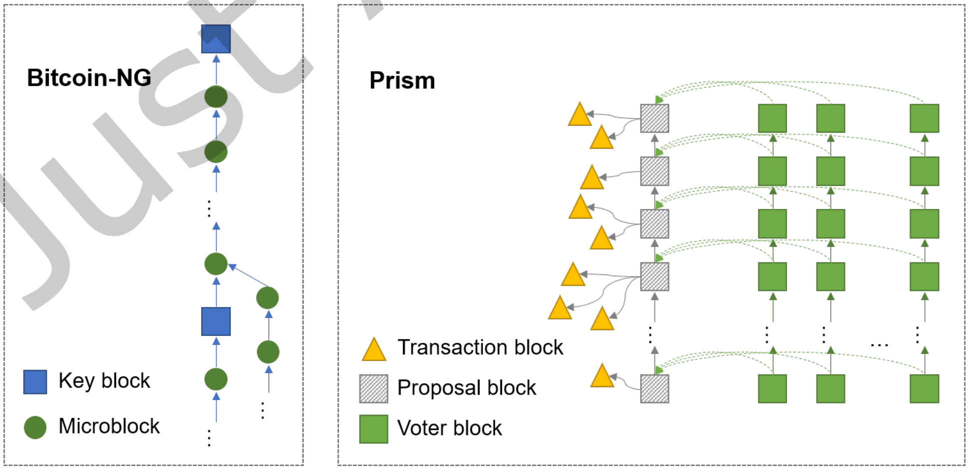

Fig. 1. Overview of decoupling functions of Bitcoin-NG [49] and Prism [10]

As a follow-up work, Prism [10] claims that its throughput can reach the network physical limits with decoupling functions. It decouples functions of blocks into transaction blocks for transactions packing, proposal blocks for leader election, and voter blocks for block confirmation. As shown in Figure 1, there is one proposal chain composed of proposal blocks and m voter chains composed of voter blocks. Each proposal block not only points to its previous proposal block, but also to multiple transaction blocks. Each voter block points not only to the previous voter block, but also to a proposal block at the same level. All miners mine to extend all the blockchains simultaneously. The input of the puzzle is $m + 2$ possible blocks instead of a transaction Merkle tree in Bitcoin, including one transaction block, one proposer block, and m voter blocks of different voter chains. The mining difficulty is reduced so that more types of valid blocks can be generated and broadcast in parallel. Once a puzzle is solved, the hash value of the puzzle determines the type of the block. The miner discards unnecessary information and broadcasts the block. When a proposal block is appended to the proposal chain, the transaction blocks it points to are also appended to the chain. This increases Prism's throughput to more than 7000 transactions per second in the network of 1000 EC2 Virtual Machines with the bandwidth of 400 Mbps [126]. Block confirmation latency is decreased to tens of seconds [126] in the above-mentioned experimental environment because m voter blocks confirm the proposed block in parallel.

NC-Max [133] decouples transaction synchronization and confirmation, and pipelines the synchronization of fresh transactions and the confirmation of non-fresh transactions to reduce fresh transactions. Fresh transactions are transactions that have just or have not yet started propagating to the network but have been included in a block. Nodes' querying for fresh transactions increases the time to verify new blocks, resulting in longer block propagation time and lower blockchain throughput. NC-Max accelerates block propagation based on the compact block mechanism [33], thereby increasing blockchain throughput.

4.2.2 Parallel Chains. It is a common idea to allow miners to extend parallel chains simultaneously to improve the blockchain throughput, not only in Prism, but also in [90, 121, 129]. OHIE [129] is a representative protocol for parallel-chain schemes. It is a k-chain structure consisting of k parallel instances that implement Nakamoto consensus protocol. The genesis block of each instance is independent of each other. All miners compete to solve puzzles. The Merkle root of a block is computed using the last block of each chain rather than the previous block as the input to the puzzle. The random hash value of the puzzle solution determines which of the k chains the generated block is appended to. Therefore, this mechanism forces computing power to evenly split across k chains, and adversaries cannot concentrate on attacking one of the chains. In order to serialize the blocks on k chains, there is an additional tuple *(rank, next_rank)* in each block. Blocks are sorted according to the increasing *rank* value.

Before OHIE, Chainweb [90] is another parallel-chain effort using PoW. Each block header contains the Merkle root computed with blocks of other peer chains, thus forming a more compact structure. Unfortunately, Chainweb only considers some specific attack strategies, and cannot resist that adversary concentrating computing power to attack one of the sub-chain. Kiffer *et al.* [70] proposed a Markov-chain-based blockchain consistency analyzing method, and found that Chainweb cannot achieve its claimed throughput of $10K$ transactions per second. Kiffer *et al.* proposed CliqueChain, a variation of the Chainweb. They proved that if CliqueChain achieves the same consistency as Nakamoto consensus protocol, it can only achieve the same throughput.

Most parallel-chain protocols do not consider workloads division. Nodes have to replicate the communication, storage, and state of the whole network. Monoxide [121] divides workload including communication, computing, and storage into multiple independent and parallel consensus zones, one blockchain for each zone. Nodes reach consensus with smaller communications in each zone. Partition causes the computing power of honest nodes to be split. In order to prevent adversaries from attacking a zone by concentrating their computing power, Monoxide allows miners to generate multiple blocks appended to multiple zones by solving a PoW puzzle. This mining mechanism amplifies the effective mining power of honest nodes and distributes their mining power evenly in various zones. Thus, the ratio of the computing power between the honest nodes and adversaries in each zone is the same as that of the whole network.

Wang *et al.* pointed out that existing parallel-chain schemes assume that the mining power is constant [122]. This assumption is not in line with practice. In fact, mining power always changes dynamically. In order to adapt to the dynamic change of mining power, the mining difficulty of parallel chains also needs to be adjusted dynamically like Bitcoin to maintain blockchain security. Wang *et al.* [122] proposed a general and provably secure methodology for parallel-chain mining difficulty dynamic adjustment. One of the parallel chains is used as the pivot chain to dynamically adjust the mining difficulty. Other non-pivot chains must refer to the pivot chain and their mining difficulty is non-decreasing. The concept of level in the original parallel-chain scheme is replaced by the difficulty of the chain to ensure liveness. It is feasible to use the pivot chain to set the difficulty target of all blocks, but the adaptability is poor. Using information from all chains to determine the difficulty target can achieve greater adaptability, which is a potential research direction.

4.2.3 DAG-based Protocols. Blockchain is not limited to a simple single chain or parallel chains, but a tree or Directed Acyclic Graph (DAG). DAG-based protocols also allow concurrent block generation. Unlike parallel-chain protocols with multiple independent genesis blocks at initialization, there is only one genesis block in DAG-based protocols. If the DAG-based blockchain adopts the longest chain rule, there will only be one longest chain instead of multiple chains in the parallel-chain scheme. Blocks on the forks will be pruned, and blockchain throughput cannot be improved.

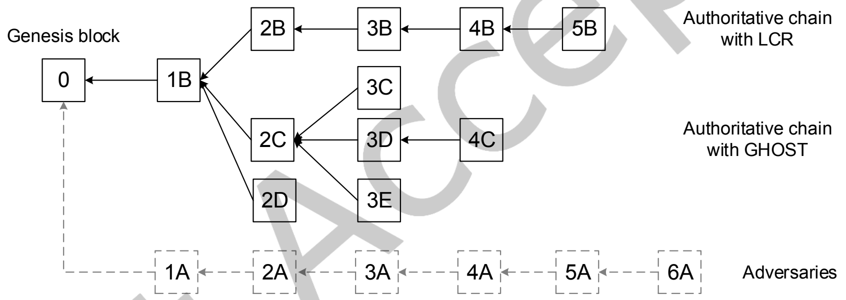

Fig. 2. Comparison of authoritative chains selected by LCR and GHOST.

GHOST (Greedy Heaviest-Observed Sub-Tree) [114] aims to enhance the security of the blockchain with high throughput. The solution of the GHOST is to follow the chain with the heaviest subtree when the blockchain is forked, instead of following the longest chain in Nakamoto consensus protocol. All the uncle blocks and the child blocks of the uncle blocks are counted when comparing the votes for the PoW obtained by the ancestor block. As shown in Figure 2, the heaviest subtree is 0, $1B$, $2C$, $3D$, $4C$. According to GHOST, the blockchain ending with $4C$ is considered authoritative, and miners choose to continue mining afterward. If adversaries publish a blockchain ending with $6A$, miners will switch to $6A$ to continue mining according to LCR, but will not switch according to GHOST. Ethereum [23] implements an improved GHOST protocol. The nephew block containing the uncle block gets some additional mining reward, while the uncle block gets a partial mining reward. This incentive mechanism promotes Ethereum blockchain convergence.

Global transaction serialization is a more complicated issue in the DAG-based protocol. In parallel-chain protocols, the number of valid blocks in each epoch/level is constant and known, so a straightforward method is to sort the blocks of each epoch/level according to the value of the chain identifier. However, there is no chain identifier in the DAG-based blockchain, so global transaction serialization is a challenging problem. Spectre [113] and Phantom [85] are two studies involving the global serialization of DAG-based blockchain. Spectre [113] not only allows a parent block to have multiple child blocks, but also allows a child block to have multiple parent blocks. Miners use the hash values of all blocks at the end of the blockchain that it knows as input for mining. Spectre confirms transactions through a recursive voting algorithm. However, the block confirmation mechanism cannot quickly achieve the global serialization of a large number of transactions. Authors of Spectre followed up with a new DAG-based protocol called Phantom [85]. In Phantom, global transaction serialization is a Maximum K-Cluster SubDAG problem, which is an NP-hard problem. Phantom proposes an optimal solution that uses a greedy algorithm to select blocks in DAG. But Phantom does not provide a formal security proof, and Li *et al.* [83] pointed out that Phantom is vulnerable to liveness attacks.

Li *et al.* [83] proposed Conflux, which processes concurrent transactions optimistically and defers the global transaction serialization. A block contains not only the hash of its parent block, which is denoted as the parent edge, but also the hashes of all known blocks of the blockchain that are not pointed to, which are denoted as reference edges. Conflux uses parent edges and the GHOST rule to determine the pivot chain and divide the DAG into multiple epochs. Then, blocks in each epoch are sorted by the relationship of reference edges. Blockchains that directly adopt GHOST are vulnerable to liveness attacks [83]. Conflux introduces a conservative strategy for resisting liveness attacks, and an optimistic strategy for high throughput and fast transaction confirmation. These two strategies can be switched according to an adaptive weight mechanism. Therefore, Conflux breaks the throughput bottleneck of the blockchain while ensuring liveness. However, Conflux does not consider actual transaction demands. Even when the transaction demand in the network is low, a large number of empty blocks may continue to be generated, which causes a waste of computing power and communication [125]. Occam [125] is built on a structured DAG that can expand or shrink exponentially. Miners judge the current transaction demand by the size of the blocks that have been appended to the blockchain. When the transaction demand is high, the mining difficulty decreases adaptively and Occam scales up; when the transaction demand is low, the mining difficulty increases adaptively and Occam scales down. The mining power load balancing mechanism prevents adversaries from concentrating computing power to attack a sub-chain, ensuring the security of the consensus protocol.

4.3 Protocols for Improving Decentralization

PoW mining is an incentive-driven activity, which encourages competitive miners to use high-performance computers for mining. The incentive mechanism of Bitcoin will lead to system centralization [48, 78, 102]. As more nodes participate in the blockchain, the requirements for computing power that can be rewarded are getting higher, which is not conducive to individual participation. Bitcoin miners tend to participate in mining pools to obtain more stable profits. There are two main directions to improve the decentralization of blockchain. One is to prevent nodes from participating in centralized mining, and the other is to promote more decentralized nodes to join the mining.

Miller *et al.* [92] proposed a direct method to alleviate Bitcoin centralization, that is, to disrupt the reward distribution of mining pools. The original PoW puzzle is replaced by the strongly nonoutsourceable puzzles to prevent mining coalitions. The scheme can guarantee that if the mining pool outsources strongly nonoutsourceable puzzles to miners, miners can steal rewards without being caught.

Lynx [88] adopts another design strategy. Its consensus algorithm is Hybrid Proof of Work (HPoW), which is a PoW variant that is more friendly to low-power devices. The incentive mechanism of Lynx is not deterministic but random. Even the fastest miner cannot guarantee rewards. It is sustainable and profitless to encourage more individual miners to participate in consensus.

The equipment involved in mining has become more energy-efficient and professional. The computing power generated by Application-Specific Integrated Circuits (ASICs) has accounted for the main computing power of

Bitcoin [100]. This trend hinders the joining of ordinary nodes and aggravates the centralization of the blockchain system. There are many consensus protocols whose goal is to reduce the efficiency gap between special-purpose computing equipment and general personal computers as much as possible. Most consensus protocols that prevent ASIC-based miners from gaining the absolute advantage in mining use memory-hard puzzles [20]. The main design idea is to increase the memory and bandwidth requirements to solve the puzzle, so as to resist ASICs using memory in parallel to find nonce [123]. The hash function used by Bitcoin is not a memory-hard puzzle, which is easy for ASIC to find solutions [123]. Litecoin [86] and DogeCoin [41] use Scrypt [104], the most popular memory-hard puzzle, to provide opportunities for CPU and GPU-based miners to participate in the competition. Others such as X11 [43] used in DASH [35] contains 11 SHA3 that complicate the effective implementation of ASICs. ETHash [123] used by Ethereum [47], Equihash [15] used by Zcash [131], CryptoNight used by Monero [95] are also ASIC-resistant, and friendly to equipment with CPU.

4.4 Protocols for Improving Security

The improper design of blockchain consensus protocols may lead to various security attacks, such as selfish mining attacks [50], double-spending attacks [132], liveness attacks [83], etc.

Bitcoin assumes that miners broadcast the mined block immediately after finding it. Nodes that control X percent of computing resources are expected to get X percent rewards. However, malicious adversaries can delay the release of mined blocks, resulting in a waste of the computing power of honest nodes. This is the selfish mining attack [50]. Adversaries can obtain higher rewards than normal mining by selfish mining attacks if they control more than 1/3 of the computing power. In order to resist selfish mining attacks, Pass *et al.* proposed Fruitchains [102], which contains a more effective incentive mechanism to achieve a fair blockchain system. They defined a new data structure called fruit. Transactions are included in fruits, and fruits are included in blocks. Miners can mine a fruit and a block at the same time by performing a hash operation. The prefix of the puzzle hash determines whether the fruit mining is successful, and the suffix determines whether the block mining is successful. The mining difficulty of generating fruit is low, which encourages more independent nodes to join. Nodes get rewards if they successfully generate fruits. Fruitchain requires the fruits in blocks to be ‘recent’, so that the adversary cannot delay the release of the mined block to initiate a selfish mining attack.

Although the PoW algorithm expects to generate a new block every ten minutes, in fact the variance of the block generation intervals is high [16]. The unstable block generation rate makes Bitcoin more vulnerable to double-spending attacks and selfish mining attacks [50]. In Bobtail [16], miners keep finding random numbers until the average of the k lowest random numbers is less than the threshold. The miner who finds the smallest random number determines the transaction in the new block, while all miners who contribute are rewarded. The lower variance mining and the reward scheme help to enhance the security and robustness of the consensus protocol.

StrongChain [117] modifies the Bitcoin consensus algorithm and incentive mechanism to encourage miners to work collaboratively rather than compete. Strongchain not only retains the original solution of the PoW puzzle, but also introduces weak solutions. Weak solutions are not enough to solve the PoW puzzle but still contribute. Miners with weak solutions can also get rewards. Miners who completely solve the original puzzle need to embed weak solutions. By embedding weak solutions, a block can aggregate more computing power, so it can better measure the computing power that contributes to this branch. Adversaries will take greater risks if they launch selfish mining attacks, and double-spending attacks also require more computing power.

PoW-based consensus protocols are energy-consuming, and their performance is unsatisfactory. Instead, consensus protocols based on Proof of Stake (PoS) and Proof of Storage (PoStorage) are more environmentally-friendly and efficient. Nodes can participate in consensus as long as they hold stakes or storage.

Table 2. Comparison of PoS-based consensus protocols

Schemes	**Methods**	**Comments**
Peercoin [72]	Coin age consuming & Checkpointing mechanism	Require PoW
Nxt [32]	Target value & cumulative difficulty	Highest cumulative difficulty
Snow White [14]	Limit cryptocurrency liquidity	Formal PoS consensus model
Ouroboros [69]	Secure MPC coin-flipping for leader election	Allow stake delegation; synchronous
Ouroboros Praos [36]	VRF & forward secure signature	Build on Ouroboros; partially synchronous
Ouroboros Genesis [9]	Bootstrap from genesis	Full dynamic availability; without checkpointing
Ouroboros crypsinous [68]	SNARKs coin evolution & key forward secure encryption	PoS privacy preserving
Casper [24]	Checkpoint & deposit penalty	Justified checkpoint of the greatest height instead of LCR
Algorand [57]	Non-interactive VRF & improved BA	Secure participant replacement
LaKSA [106]	Votes recorded in blocks	Most-stake rule instead of LCR; favors availability over consistency

5 POS-BASED BLOCKCHAIN CONSENSUS PROTOCOLS

The PoS-based consensus protocol is considered to be a promising alternative to PoW due to its energy-saving property. Leaders for generating new blocks are selected by a cryptographic random algorithm rather than solving puzzles. The probability of being selected depends on how many stakes the node has. In general, consensus protocols based on PoS can be classified into two types [24]. The first type is the chain-based PoS protocol, such as Peercoin [72], Nxt [32]. The second type is BFT-based PoS protocol, such as Algorand [57] and Tendermint [77]. The comparison of PoS-based consensus protocols is summarized in Table 2.

5.1 Chain-based PoS

Chain-based PoS consensus protocols are similar to Nakamoto consensus protocol. In each round, nodes execute a random algorithm to determine whether they are selected as the leader. The leader generates a new block and broadcasts it. Nodes append the received block to the end of the blockchain. Multiple blocks may be appended to one block due to network delay or malicious attacks. Chain-based PoS protocols also have specific rules, such as the most-stakes chain, to determine the authoritative chain to ensure the consistency of the blockchain.

Peercoin [72] is a milestone in the development of PoS. It is the first large-scale project based on a hybrid consensus of PoW and PoS. It contains both blocks generated by PoW and blocks generated by PoS. Coin age is the core of PoS block generation, and is defined as the units of coins multiplied by the holding period. For example, if a node has 10 coins and has held it for 90 days, the node has accumulated the coin age of 900 coin days. If the coin is transferred, the coin age is consumed. Coin age consumption is the stake in Peercoin. Nodes find solutions in the limited search space. If a solution satisfies a target, a new block can be generated. The target is determined by the coin age consumed. Thus, more coin age consumption means a higher block generation priority. Peercoin nodes do not follow LCR. It chooses the fork with the highest coin age as the authoritative chain, and the coin age consumption of all transactions contained on it is be counted. Even if adversaries control more than half of the computing power, it does not mean that they can attack Peercoin successfully. If an adversary attempts to launch the double-spending attack, his coin age will be consumed and it will take a long time to recover. Peercoin still requires proof of work, and it has not fully achieved the goal of reducing energy consumption. Nxt [32] is the first pure PoS cryptocurrency. The more cryptocurrencies a node has, the greater the chance to get the right to generate blocks.

Bentov *et al.* [14] formally defined the functionalities and robustness requirements of the PoS consensus protocol. Even if network jitters, nodes join or exit frequently, the consensus protocol can ensure consistency and liveness. They also proposed an example 'Snow White' that meets the requirements based on their previous work named sleepy model [103]. Snow White restricts cryptocurrency liquidity to defend against double-spending attacks. A new committee is selected in each epoch based on the parameters of the blocks that have been appended to the blockchain. At every step in an epoch, a leader is selected in this committee by hash operation. The leader generates a new block and broadcasts it.

Ouroboros is a series of works based on PoS. Unpredictable and unmanipulable election of leaders is especially important in PoS-based consensus protocols as stakes are transferred and the number of stakeholders changes. Common solutions include randomly electing a leader utilizing a pseudo-random function, and determining the next leader based on the existing blockchain state. Kiayias *et al.* [69] proposed Ouroboros, which distinguishes from those solutions. In each epoch, a group of stakeholders executes a secure multi-party coin flip protocol to select stakeholders and leaders for the next epoch randomly. The leader generates a new block for the epoch. It provides a formal model on the persistence and liveness of PoS-based blockchain protocols. The authors rigorously proved that Ouroboros is secure under certain assumptions, such as that the network is synchronous, stakeholders are online. They also introduce a stake delegation mechanism applicable to Ouroboros. Through proxy signature, stakeholders can participate in consensus and generate new blocks without personal participation.

Ouroboros Praos [36] is an improved protocol based on Ouroboros [69]. Ouroboros Praos utilizes a special Verifiable Random Function (VRF) for leader election, which makes the protocol unpredictable even if an adversary compromises the key generation process. Compared with Ouroboros, it can resist desynchronization attacks, and is suitable for the partially synchronous network. Dembo *et al.* [38] provides the security analysis of Ouroboros Praos using their proposed method of turning attacks into a race between the adversary and honest nodes.

The chain-based PoS protocols are difficult to compare with PoW in terms of achieving full dynamic availability. Full dynamic availability means that as long as the computing power of honest nodes accounts for the majority, the blockchain security is not affected by any node going online or offline [37]. Ouroboros Genesis [9] aims to enhance the full dynamic availability of the PoS protocol. As follow-up work, its structure is similar to Ouroboros Praos [36], but the chain selection rule is local rather than the original global longest chain rule. It allows nodes to join the consensus protocol based on the genesis block, without the need for checkpoints. Thus, Ouroboros Genesis resists adaptive adversaries, and nodes can join or exit arbitrarily without external trusted parties.

Privacy preservation is a challenging work for PoS-based protocol because of open and queryable stakes. Ouroboros crypsinous [68] combines the advantages of Zerocash [108] and previous Ouroboros series work to preserve the anonymity of nodes during the transaction and block proposal process. However, recently Kohlweiss *et al.* [73] formally proved that by leveraging network delay, the identity of the block proposers in Ouroboros crypsinous can be revealed even under ideal anonymous broadcast settings. They show that if adversaries are able to control the network delay, it is impossible to design a PoS-based consensus protocol that guarantees both liveness and anonymity. In order to improve the interoperability, scalability and upgradability of Ouroboros series protocols, Gaži *et al.* [56] proposed a PoS sidechain construction. The sidechain ensures the safety of cross-chain value transfer when both chains meet the security assumptions. It also maintains firewall properties, that is, the collapse of one chain will not damage the other chain. The sidechain with 2-way pegs [8, 112] enables different types of blockchains to communicate, realizing the diversification and specialization of the blockchain system. By transferring all assets to the updated sidechain and then discarding the mainchain, the sidechain can become the updated mainchain.

Deb *et al.* [37] pointed out that most PoS-based protocols generally need additional assumptions: all adversaries have existed since the genesis block and no new adversary participates afterward [37]. This assumption is too strong and not practical for real-world systems. They proposed PoSAT that uses Verifiable Delay Function (VDF) to avoid the additional security assumptions and achieve the same full dynamic availability as PoW. In order to generate a new block, the node needs to iteratively compute VDF with the randomness of the parent block and the public key, until it finds a value that is less than the threshold. Adversaries need to spend more time extending their forked chains. Therefore, it is difficult for adversaries to catch up with the honest and longest chain.

5.2 BFT-based PoS

BFT-based PoS blockchain consensus protocols are hybrid protocols based on both vote and proof. First, several validators (including a special validator called the leader) are selected to form a committee. Common selection methods are: deposit locking in advance, such as [24, 119]; cryptographic selection based on account balance, such as [57, 106]. Then, the leader proposes a new block. Committee implements classic BFT consensus algorithm and decides whether the new block is valid through voting.

Casper [24] is proposed to be an overlay on top of the Ethereum block generation layer. The Ethereum block generation layer is a PoW-based blockchain with many forks, which is actually a blocktree. Casper aims to generate a canonical chain for the block generation layer to maintain the single-chain structure. The canonical chain is based on the PoS protocol and composed of a series of justified checkpoints. Starting from the genesis block, the checkpoints are the blocks on the blocktree whose height is a multiple of 100. The checkpoints are justified by validators voting. Nodes become validators by depositing in blocks before the new checkpoint. Once a validator violates rules, the penalty is deducted from its deposit. The weight of the validators' votes is proportional to their deposit. Only when the checkpoint gets more than 2/3 votes, the checkpoint is justified. The canonical chain is from the genesis block to the justified checkpoint. Nodes follow the chain with 'the justified checkpoint of the greatest height' when the blockchain forks.

Nodes in Tendermint [119] lock deposits in bond transactions to become validators. Validators take turns to become proposers to propose blocks, and there is only one proposer in each round. Before the new block is committed, validators perform the two-phase commit, including prevote and precommit. Only after receiving more than 2/3 of the votes in the previous phase, the protocol will proceed to the next phase. In order to prevent the blockchain from forking, Tendermint introduced a locking mechanism. If a validator has prevoted for a block in some prior round, the validator will be locked in this block. Validators cannot vote for a different block in another round before it is unlocked. The validator will unlock only if a block in the following round receives more than 2/3 votes.

Algorand [57] does not set a deposit penalty mechanism but utilizes the Verifiable Random Functions (VRF) to select leaders and validators. VRF is a non-interactive algorithm. Nodes run it locally and get the selection result. The input of VRF contains the amount of cryptocurrency owned by the node. The more cryptocurrency a node has, the greater the probability of being selected. The leader and validators implement an improved Byzantine agreement (BA) called BA★ to reach consensus. Most schemes only select leaders and validators once in one round of consensus. However, in Algorand, the leader and validators in a round of BA★ are selected not only once. After receiving the results of the previous step in a round of consensus, the VRF re-elects validators to proceed to the next step. More specifically, after receiving the block of the previous epoch, the node runs VRF to determine whether it is the leader. If it is, it generates a new block and broadcasts it. After receiving the new block, all nodes run VRF again to determine whether they can vote as a validator. Validators validate the block

Table 3. Comparison of PoStorage-based consensus protocols

Schemes	Consensus	Aims	Methods	Comments
Filecoin [51]	PoRep	Data storage	Challenge/response	Only valid during challenge/response
Filecoin [51]	PoSt	Data storage time	Iteratively execution	Reduce communication costs
SpaceMint [100]	PoSpace	Dedicate disk space	Pebbling game & block quality	Non-interactive
Chia [30]	PoSpace	Dedicate disk space	VDF	No synchronization requirement
Permacoin [91]	PoR	Distributed storage	PoR lottery	Non-interactive

and broadcast their votes. After receiving the previous vote, the validators are selected again to vote for the next step. The validator selection and voting are repeated until enough nodes in the committee reach consensus. Since the leader and validators in BA⋆ are re-selected at each step, Algorand can resist targeted attacks caused by identity exposure.

LaKSA [106] uses a cryptographic sampling algorithm in each round to select committee members. The algorithm creates an array containing all stake units and randomly samples some units. The probability of being selected is proportional to the number of stakes the node has. The validators validate and vote on the block generated by the leader. If the block receives enough votes, the block is confirmed. LaKSA prefers availability rather than consistency, which is different from most consensus protocols. The votes of the committee members are recorded in blocks so that nodes can calculate the probability of the specific block being reverted. Each node can personalize its security thresholds. Block confirmation is determined by each node individually. LaKSA absorbs the design idea of GHOST and follows the most-stake rule to deal with blockchain forks.

6 POSTORAGE AND POX-BASED BLOCKCHAIN CONSENSUS PROTOCOLS

6.1 PoStorage

Proof of Storage (PoStorage) is a classic technique used in file storage systems [5, 22, 65]. The basic idea is that the storage server can prove the integrity of the files that the user can access. PoStorage can also be used in blockchain to identify Sybil nodes. Miners generate proofs with their memory, disk space, etc. There are many variants based on PoStorage, including Proof of Replication (PoRep) [51], Proof of Spacetime (PoSt) [51], Proof of Space (PoSpace) [45], Proof of Retrievability (PoR) [22, 65], and Proof of Capacity etc. The comparison of PoStorage-based consensus protocols is summarized in Table 3.

6.1.1 Proof of Replication & Proof of Spacetime. Filecoin [51] is a blockchain for file storage, which involves both Proof of Replication and Proof of Spacetime. Miners are storage providers. Users pay to store their files on miners' physical storage devices. The reward for miners is proportional to the amount of storage they contribute to Filecoin. Proof of Replication is used to prove that the data is received and uniquely stored on a dedicated physical device. Proof of Spacetime is used to prove that the data is stored during a specified period of time.

PoRep is a challenge/response protocol. First, a miner sets up and stores a pseudo-random permutation of data based on the miner's public key, and returns some corresponding information to the user. The user initiates a random challenge based on this information. Then, the miner generates a response of corresponding data storage proof, which is a proof of zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs). The miner returns it to the user for validation.

However, PoRep has a flaw [13]. It can only guarantee that miners uniquely store certain data during the challenge/response period. Proof of Spacetime requires miners to execute the required number of PoRep iteratively. The output of the previous PoRep is used as the input for the next time. The miner returns the proof of PoSt for validation and gets rewards. Therefore, PoSt guarantees that miners store certain data throughout a period of time. PoSt avoids the communication costs caused by multiple repeated challenges/responses.

6.1.2 Proof of Space. SpaceMint [100] is a blockchain project based on proof of space proposed by [45, 107]. It modified the original PoSpace protocol [45] to be non-interactive. In the initialization phase, miners broadcast their space commitment. Commitments including space, coin transfer records, and punishments for malicious behavior are recorded on the blockchain. Miners will be punished if they fail to contribute disk as committed. In the mining phase, miners generate proof with the state parameters of the blockchain. The proof is based on a cryptography game called *pebbling*. Pebbling games can guarantee that the proof can be generated when the data is stored correctly. The more disk space the miner invested, the higher the quality of the proof. The miner generating the highest quality proof becomes the new leader. All miners always follow the blockchain with the higher block quality. Chia [30] is another blockchain project based on PoSpace. It is implemented with the VDF [18, 19] to alleviate long-range attacks.

6.1.3 Proof of Retrievability (PoR). Proof of Retrievability [22, 65] allows validators to regularly check the retrievability of files stored by the prover. The file is encoded and embedded with error correction codes and random values for checking, named *sentinel*. The validator initiates a challenge and requires the prover to respond to the value of the sentinel at the specified location. If the prover has modified or deleted the file, it is difficult to respond correctly. Permacoin [91] uses a non-interactive PoR lottery instead of mining to build a distributed file storage.

6.2 Proof of X

According to the requirements of different scenarios, countless consensus protocols that rely on different resources as proofs can be designed. We collectively call it *'Proof of X'*, including Proof of Elapsed Time, Proof of Meaningful Work, Proof of Activity, Proof of Work time, Proof of History, etc.

6.2.1 Proof of Elapsed Time (PoET). PoET adopted by Hyperledger Sawtooth [109] relies on the Trusted Execution Environment (TEE) to ensure confidentiality, integrity, and randomness. There is no mining process in PoET. The mining interval is simulated by a timer function in the enclave. In order to generate a new block, each node requests a waiting time from its enclave. The node with the shortest waiting time becomes the leader. The enclave generates a signed waitCertificate for authentication. The probability of a node being selected as the leader is proportional to the number of processors with the TEE. However, Wang *et al.* [120] discovered Multi-Certificate Attack, which allows adversaries to generate multiple certificates against PoET to increase the probability of winning the leader election. So they present PoETA to resist sequential multiple certificate attacks in PoET.

6.2.2 Proof of Meaningful Work (PoMW). It is a waste of computing power and not environmentally friendly to prove that nodes are not Sybil nodes. Researchers are committed to making full use of these computing resources to do some more meaningful work. One basic idea is to replace the original computation for finding a puzzle solution with more useful scientific research problems. New and meaningful problems must be generated automatically, continuously and infinitely, and the solutions of the puzzles must be easy to validate, etc. The problem of Primecoin [71] is to find a sequence of large prime numbers. Large prime numbers are very useful security parameters in cryptographic algorithms. Proof of eXercise [111] requires miners to solve a scientific computation matrix-based problem, which contributes to image recognition and data mining research.

7 COMMITTEE-BASED CONSENSUS PROTOCOLS

Voting is a classic method to reach consensus in the distributed system. The basic idea of committee-based consensus protocols is that if the number of votes in favor of a proposal exceeds the threshold, the consensus is reached. Committee-based protocols can directly reach deterministic consensus after multiple rounds of interaction. It means the new block appended to the blockchain will never be replaced. Some classic committee-based protocols can only be applied in the permissioned blockchain [26, 42, 60, 93] since more nodes lead to low efficiency. Recently, more efficient committee-based consensus protocols for permissionless blockchain have also developed rapidly [74, 75, 87, 119]. Next, we discuss two categories: committee-based permissioned blockchain consensus protocols and committee-based permissionless blockchain consensus protocols.

7.1 Committee-based Permissioned Blockchain

Some consensus protocols are designed for distributed systems with a limited and known number of nodes, such as Paxos [79], and Raft [99]. They are simple CFT consensus protocols that cannot tolerate Byzantine faults and are not suitable for blockchain. Many researches [26, 60, 93] are devoted to making the consensus protocol tolerant Byzantine faults. These BFT protocols [26, 60, 93] were originally designed to ensure that the non-faulty nodes agree on the execution order of the service commands initiated by the client even if there are f Byzantine nodes [128]. However, they are also suitable solutions for building a permissioned blockchain to collectively maintain an ordered ledger [42, 128].

In this section, we first introduce Practical Byzantine Fault Tolerance (PBFT) [26], which was proposed to work in asynchronous environments such as the Internet. Consistency does not depend on network synchronization, but liveness still depends on network synchronization. Then, we introduce HoneyBadgerBFT [93]. HoneyBadgerBFT is the first practical asynchronous BFT protocol, and its security does not depend on the time of message delivery. Finally, we introduce some research on improving the throughput of the BFT protocol, reducing communication costs, enhancing fairness and flexibility, etc. The comparison of committee-based permissioned blockchain is summarized in Table 4.

7.1.1 Practical Byzantine Fault Tolerance. PBFT is built on the state machine replication model in distributed systems. The purpose of the protocol is that multiple nodes respond to requests from the client and execute operations in the same order [26]. It adopts primary-backup techniques to order requests. The protocol execution proceeds in views. There is one and only one node in a view that is the primary, and the other nodes are backups. The primary can be regarded as the leader, and the backups are validators when PBFT is applied in the blockchain.

As shown in Figure 3, the normal process of PBFT includes the following phases:

- Request: The client sends a request message that includes an operation, a timestamp, and an identifier to the leader.
- Pre-prepare phase: The leader first generates a sequence number for the request message. Then, the leader signs the current view number, the sequence number, and the digest of the message. Finally, the leader multicasts to other validators a pre-prepare message containing the signature and the message from the client.
- Prepare phase: Validators first check the signature, view, and sequence number in the pre-prepare message. If they are all valid, the validator starts the prepare phase. It signs the digest of the pre-prepared message, view, sequence number and its identifier. Then, the validator takes the signature as the prepare message, stores it locally and multicasts it to other validators.
- Commit phase: If a validator receives more than $2f$ the prepare messages matching its pre-prepare message and the content are valid, it starts the commit phase. The leader and validators multicast commit messages,

[1] γ is the order-fairness parameter, and $\frac{1}{2} < \gamma \leq 1$

Table 4. Comparison of committee-based consensus protocols

Schemes	Network Model	Fault Tolerance	BFT Corruption Threshold
Paxos [79]	Synchronous	CFT	Not applicable
Raft [99]	Synchronous	CFT	Not applicable
Sync HotStuff [2]	Synchronous	BFT	$n \geq 2f + 1$
PILI [29]	Synchronous	BFT	$n \geq 2f + 1$
PBFT [26]	Partially synchronous	BFT	$n \geq 3f + 1$
HotStuff [128]	Partially synchronous	BFT	$n \geq 3f + 1$
PALA [28]	Partially synchronous	BFT	$n \geq 3f + 1$
HoneyBadgerBFT [93]	Asynchronous	BFT	$n \geq 3f + 1$
BEAT [42]	Asynchronous	BFT	$n \geq 3f + 1$
Dumbo [60]	Asynchronous	BFT	$n \geq 3f + 1$
Byzcoin [74]	Partially synchronous	BFT	$n \geq 3f + 2$; Less than 1/4 hash power
Flexible BFT [89]	Synchrony & Partial synchrony	BFT	$n \geq 2f + 1$ for synchrony; $n \geq 3f + 1$ for partial synchrony
Pompē [134]	Partially synchronous	BFT	$n \geq 3f + 1$
Aequitas [67]	Synchronous & Asynchronous	BFT	$n \geq 2f/(2\gamma - 1)$ for synchrony; $n \geq 4f/(2\gamma - 1)$ for asynchrony[1]

which contain a signature of the view, sequence number, and its identifier. If the leader or validator receives more than $2f + 1$ valid commit messages, the commit phase is completed. Each validator executes the requested operation and returns the result to the client.

- Replay phase: If the client receives $f + 1$ reply messages with the same result, it is considered that a consensus has been reached.

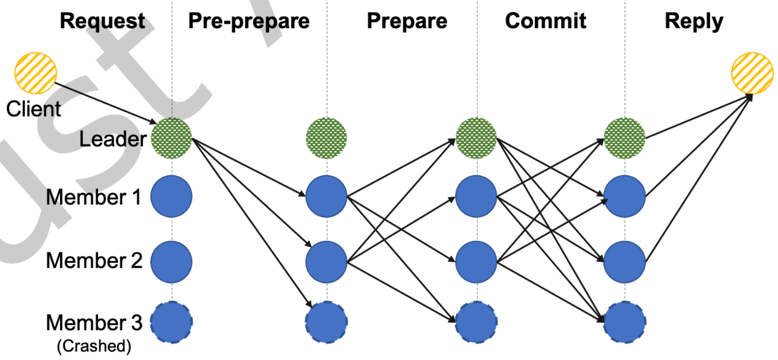

Fig. 3. The normal consensus process of PBFT [26].

The above is the normal PBFT consensus process. PBFT can ensure consistency if there are a total of $3f + 1$ nodes and fewer than f Byzantine faulty. But it cannot handle the malicious leader. To address this case, PBFT includes a view change mechanism, which is triggered when a validator thinks that the current view cannot make progress. Each validator starts a timer when it is waiting for messages. The timer stops when the validator is not waiting, and restarts when the validator receives a new request again. If it times out, the validator broadcasts a view-change message. Once the leader of the next view receives more than $2f + 1$ view-change messages, it broadcasts the new-view message to validators. After validators validate the new-view message, they return to the normal PBFT consensus process again.

There are many subsequent blockchain projects based on PBFT. For example, ByzCoin [74] has improved the communication costs of PBFT with a collective signing protocol and couples a PoW blockchain to make ByzCoin suitable for permissionless environments. Many blockchain projects [75, 87] combine sharding and PBFT to design blockchains for large-scale networks. Tendermint [77] also achieves the final consensus through prepare and commit voting, but improves leader switching.

7.1.2 HoneyBadgerBFT. Prior to HoneyBadgerBFT [93], BFT protocols in permissioned environments, such as PBFT, are based on synchronous or partially synchronous network assumptions. Miller *et al.* constructed an 'intermittent synchronization' network to prove that protocols based on a partially synchronous network would lose liveness in this case. This means that adversaries can make the throughput of these protocols zero. It is important for consensus protocols to work properly in the asynchronous network. However, Fischer *et al.* [52] proved that no protocols can be guaranteed to reach consensus as long as there is one faulty process in a completely asynchronous system of deterministic processes, which is called FLP impossibility. Thus, researchers must make compromises to achieve consensus in asynchronous networks. For example, consensus protocols can include random beacons to make the process non-deterministic.

HoneyBadgerBFT is based on the fully asynchronous setting, which means the liveness does not rely on the assumption of message delay. All nodes broadcast messages using an improved Asynchronous Common Subset (ACS), an improved asynchronous atomic broadcast protocol based on [12]. It is a randomized agreement to avoid FLP impossibility. If each of the N nodes proposes a value, ACS guarantees that each node outputs a vector, which contains at least $N - 2f$ correct node input values. However, the throughput of using ACS directly to the blockchain consensus protocol is extremely low. In order to solve this problem, nodes randomly select transactions and broadcast the most disjoint transaction set. Transaction selection improves throughput but allows adversaries to censor transactions selectively. HoneyBadgerBFT uses threshold encryption to enhance censorship resilience. Threshold encryption prevents the adversary from knowing the transactions proposed by other nodes.

HoneyBadgerBFT algorithm consists of three phases:

- At the beginning of each epoch, each participating node randomly selects a batch of transactions from its buffer. Transactions are threshold encrypted before broadcasting.
- In the second phase, nodes broadcast transactions using ACS. Specifically, ACS includes two sub-phases: Reliable Broadcast (RBC) and Asynchronous Binary Byzantine Agreement (ABA). Transactions are broadcast through RBC. Then, ABA generates a bit vector indicating which RBCs have been completed successfully. A consistent set of transactions is formed through the ABA.
- In the third phase, each node decrypts its own part and broadcasts it. The set of transactions can be decoded after receiving at least $f + 1$ decrypted messages. Finally, these transactions are packed to generate a new block.

7.1.3 Improvements of Asynchronous BFT Consensus Protocol. BEAT [42] improves the efficiency of HoneyBadgerBFT. BEAT retains the architecture of HoneyBadgerBFT protocol. But it utilizes a more efficient threshold coin-flipping instead of the threshold encryption with great costs in HoneyBadgerBFT, which improves throughput and reduces latency. In addition, BEAT adopts a modular design and consists of five asynchronous BFT protocols to fit different applications. Functions can be selected and combined according to needs to achieve better performance. DispersedLedger [127] decouples the consensus protocol into a data availability agreement and block retrieval based on HoneyBadger. Nodes vote immediately after observing that the block has been dispersed to ensure the consistency of the consensus protocol. Then, nodes download blocks asynchronously to ensure liveness. High-bandwidth nodes no longer need to wait for low-bandwidth stragglers to download blocks and then vote, significantly improving performance under bandwidth variability.

Dumbo [60], which consists of two atomic broadcast protocols Dumbo1 and Dumbo2, takes another approach to improve HoneyBadgerBFT. Each node in HoneyBadgerBFT has to run N instances of ABA in each round of consensus, and each instance has to validate $O(N^2)$ threshold signatures. It takes a long time, far greater than the delay of RBC. Dumbo restructures the ACS protocol of HoneyBadgerBFT to improve efficiency. More specifically, Dumbo1 reduces the number of running ABA instances, thereby shortening the running time. Dumbo2 adopts the multi-valued validated Byzantine agreement (MVBA) to optimize ACS. After that, Dumbo-NG [54] and Speeding Dumbo [59] aim at a more practical asynchronous BFT consensus protocol, and further improve the performance of the protocol based on Dumbo.

7.1.4 Other Improvements of Committee-based BFT Consensus Protocol. Classic committee-based BFT consensus protocols require multiple interactions between nodes to reach consensus. Some work [2, 28, 29, 128] is dedicated to accelerating the voting process of the protocol. PILI [29] and PALA [28] are pipelined-BFT protocols under the assumption of synchronous network and partially synchronous network, respectively. In order to reduce the number of interactions, the protocols piggyback the present block's commit-vote on the next block's round. After a block is appended to the blockchain and before the block is finalized, there is a new block state called the notarized state. The notarized block still needs to wait for several blocks to be finalized. This design allows consensus on the next block even if the previous block is not finalized.

HotStuff [128] aims to simplify the complexity of leader replacement and works in the partially synchronous network. It adds a decide phase after the commit phase. The new leader can choose the highest quorum certificate he knows to drive the protocol to reach consensus. Sync HotStuff [2] is under synchronous network assumption, and no longer requires nodes to start or end a round simultaneously.

Committee-based BFT protocols [26, 42, 60, 93] first assume the network conditions, including synchronous, partially synchronous, asynchronous, the proportion of Byzantine nodes, and then design the protocol. This design idea weakens the universality and portability of protocols. For example, once the network conditions are changed, the original persistence or liveness may no longer be guaranteed. Flexible BFT [89] is a solution for improving the flexibility of the BFT protocol to adapt to networks. It deconstructs the BFT protocol and designs Flexible Byzantine Quorums. Flexible BFT allows a ledger to accommodate nodes with different assumptions, such as synchronous or asynchronous, and different proportions of Byzantine nodes. Flexible BFT relies on clients to choose the commits rule between different network assumptions. It is not secure for a client if the client chooses the commit rule of synchronous networks in the asynchronous networks. Thus, Momose *et al.* [94], proposed a multi-threshold BFT protocol that adapts to different network timing assumptions with only a single commit rule. The protocol can tolerate two-thirds of faults for safety in synchronous networks, while it can tolerate one-third of faults for liveness in synchronous networks and safety/liveness in asynchronous or partially synchronous networks.

The leader-led committee-based consensus protocols [2, 26, 99, 128] allow the leader to order transactions, but ignore the fairness of the transaction order. Malicious leaders can profit by manipulating the order of transactions. Participating nodes can only validate the transactions, and cannot prevent the leader from manipulating the order of transactions. Zhang *et al.* [134] refer to the node that controls the decision of transaction order as a Byzantine oligarchy. They proposed *Pompē* to prevent Byzantine oligarchy by decoupling command/transaction order and adding records. By associating each command/transaction with an ordering indicator, the leader can still promote efficient consensus but can no longer control the order of command/transaction. Moreover, Kelkar *et al.* [67] proposed to add a third property: *transaction order-fairness* in addition to *safety* and *liveness* to measure the fairness of protocols. They designed a consensus protocol that can guarantee these three properties called 'Aequitas'. Transactions are broadcast based on FIFO-broadcast (FIFO-BC), and then nodes agree on the transaction order using Byzantine Agreement (Set-BA). The finalization of transaction order can be determined by either the leader-based model or the leaderless model.

Successful committee-based BFT protocol forensics can identify malicious nodes. Sheng *et al.* [110] formalize the study of forensics for BFT protocols mathematically. They propose to measure the forensics support of BFT protocols with three parameters: the maximum number of Byzantine replicas under which forensic support can be provided, the number of different honest replicas' transcripts needed to guarantee proof of culpability, and the number of Byzantine replicas that can be held culpable in case of an agreement violation. They studied the forensics of BFT protocols, including PBFT [27], Hotstuff [128], VABA [3] and Algorand [57]. Their results suggest that small changes to the BFT protocol can hugely impact forensics.

7.2 Committee-based Permissionless Blockchain

Classic committee-based BFT protocols [26, 42, 60, 93] cannot be effectively used in a permissionless environment for several reasons. First, these protocols rely on the assumption that the size of the committee is known in advance and remains fixed in order to ensure safety and liveness. But in a permissionless environment, nodes can freely join or leave the network at any time, making it difficult to accurately determine the size of the committee. Second, permissionless blockchains have no entity or mechanism to manage the identities of participating nodes. This makes it easier for adversaries to launch Sybil attacks by creating multiple fake identities. Third, BFT protocols require multiple rounds of interaction to reach consensus, which can significantly increase the communication costs as the size of the committee grows. This can lead to poor throughput for large committees, as the time and resources required for consensus increase.

There are two directions for applying committee-based BFT protocols in a permissionless environment. One is to first select some nodes from the permissionless environment through a proof-based protocol, and then implement a committee-based BFT protocol among the selected nodes. In addition to the aforementioned BFT-based PoS protocols [24, 57, 119], Byzcoin [74] is a representative scheme. Byzcoin consists of two parallel chains. One is a PoW blockchain that implements Nakamoto consensus protocol. Successful miners of the PoW blockchain in the past period of time forms a committee to extend the other blockchain. Committee members implement an improved PBFT protocol. The improved PBFT protocol adopts the collective signing [116] to reduce the communication complexity to $O(logn)$. Algorand [57] argues that although the hybrid consensus protocol has successfully enabled BFT protocols to be used in permissionless settings, the use of PoW to achieve committee dynamics will lead to the blockchain forks.

The second is to combine committee-based BFT protocols with sharding, which is a classic database scale-out technique. As shown in Figure 4, the basic idea is to first partition all nodes into committees. Committees are all assigned to a different shard, and transactions also are allocated to different shards. Then, each shard executes a BFT consensus algorithm and processes transactions in parallel to generate blocks. Therefore, the storage and communication costs are greatly reduced. The blockchain throughput increases linearly with the number of participating nodes.

However, there are still some challenges to be solved in the blockchain using sharding. First, nodes need to be assigned to different committees efficiently and securely to prevent malicious nodes from clustering in one committee. Second, communication between different shards needs to ensure atomicity and isolation. A new transaction may involve previous transactions in multiple different shards, and communication between shards is unavoidable. The cross-shard transaction validation requires the collaboration of nodes in multiple shards. Distributed coordination requires significant communication costs and is vulnerable to security attacks [124].

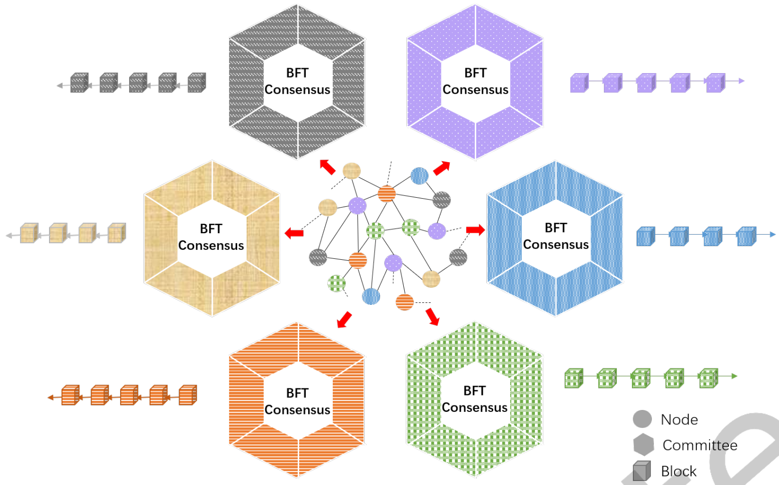

Fig. 4. Sharding architecture overview: Validators are assigned to different shards to maintain different blockchains.

Third, the BFT consensus protocol within a shard should be secure and efficient. If there are too many nodes in a shard, it will lead to high communication costs and low throughput. If there are fewer nodes in a shard but more shards, the communication costs between shards will increase, which is not conducive to performance improvement.

Elastico [87] is arguably the first blockchain consensus protocol implementing sharding. Elastico requires participating nodes to perform PoW in the first step of each epoch to resist Sybil attacks. Then, nodes are allocated to 2^s committees based on the lowest s bits of the hash result of PoW. Nodes whose hash results in the lowest s bits are the specified strings form a final committee. The other $2^s - 1$ committees are assigned to different shards to process transactions. Nodes broadcast their identities, and nodes assigned to the same committee establish connections to complete the committee setup. The intra-committee consensus can be reached through the classic PBFT. The final committee is responsible for validating and ordering the consensus results from other shards.

OmniLedger [75] pointed out that identity setup and committee formation in Elastico are insecure. Adversaries can participate in the same consensus committee by selectively publishing some of their PoW results. Moreover, the final committee is a single point bottleneck for the performance of the whole system. OmniLedger combines the VRF-based leader election in Algorand [57] and an unbiased, decentralized randomness scheme RandHound [115] to partition validators randomly into different shards. In order to support cross-shard transactions, OmniLedger designs Byzantine Shard Atomic Commit (Atomix) protocol, which contains a lock-then-unlock process. First, a user obtains proof-of-acceptance from the leader of the shard to which the input transactions belong. Proof-of-acceptance is signed by the leader of the shard, after which the input transaction will be locked. Then, the user gossips an unlock-to-commit transaction that includes all the proofs and locked transactions after he collects the proofs of all input transactions of the cross-shard transaction. Cross-chain transactions can be completed after the validation of the input transactions. Otherwise, the user gossips an unlock-to-abort transaction to abort the cross-chain transaction processing. The cross-shard transaction processing approach has some drawbacks. First of all, this user-driven approach puts a lot of burden on users, which is not conducive to lightweight clients. In addition, validators need to sign proofs for each user's request, and the communication and computing costs are very high.

Gossiping transactions to the whole network greatly increases the latency of the consensus protocol. RapidChain [130] implements a novel gossiping protocol and an optimal intra-committee consensus algorithm to improve the efficiency of consensus. RapidChain elects some nodes to form a reference committee. The reference committee is responsible for creating, configuring, and updating the committees for shards. Each subsequent epoch begins with a consensus phase followed by a reconfiguration phase. In the consensus phase, the IDA-Gossip protocol is used for rapid message propagation and validation to reach intra-committee consensus. Rapidchain validates cross-shard transactions by requiring the input committee via the inter-committee routing protocol, rather than being broadcast by the user in OmniLedger. Through these key technical improvements, the latency of Rapidchain is much lower than Elastico and OmniLedger [130] .

Unfortunately, Dang *et al.* [34] proved that Rapidchain fails to provide cross-chain transaction atomicity and isolation, and a malicious payee in OmniLedger may cause the payer's funds to be locked. Their scheme [34] uses TEE to enhance the Byzantine failure tolerance and lower communication costs. They leverage classic two-phase locking (2PL) and two-phase commit (2PC) to realize cross-chain transaction isolation and atomicity.

Cross-shard transaction processing is a major challenge for sharding-based blockchains. Some schemes, such as OptChain [97] and BrokerChain [62], aim to reduce the number of cross-shard transactions and maintain the load balance of shards by properly allocating transactions to shards. OptChain [97] designs an optimal transaction placement strategy by learning past transaction patterns, but it can only be applied to the Unspent Transaction Output (UTXO) model. BrokerChain [62] reduces cross-shard transactions based on the account model by partitioning the state graph of accounts with Metis [66]. However, BrokerChain relies on brokers with accounts in multiple shards to process cross-shard transactions one by one, which is inefficient. BrokerChain does not provide a solution for multi-input and multi-output cross-shard transactions. Tao *et al.* [118] proposed inter-shard merging and intra-shard transaction selection to reduce cross-shard communication costs. Pyramid [61] adopts a layered sharding structure, where some shards store the state of multiple shards. They expect that cross-shard transactions can be committed in a round of layered sharding consensus protocols to reduce transaction processing latency. But they failed because each phase of the layered sharding consensus protocol requires a round of consensus in another shard.

7.3 Other Miscellaneous Protocols

In addition to the mainstream consensus protocols that append blocks to the blockchain based on proof or voting, there are also some consensus protocols designed to meet the needs of special application scenarios. For example, IoT scenarios, enterprise applications, block redactable functions, etc. In this section, we first introduce transaction-based (tx-based) blockchain [64] and hashgraph [11] that do not use block aggregating transactions. Then, we introduce two industrial projects Hyperledger Fabric [4] and Delegated Proof of Stake (DPoS). Next, we introduce two non-anonymous proof-based protocols that work in the permissioned environment. Finally, we introduce some schemes [6, 39, 40] that can revoke information on the blockchain.

7.3.1 Tx-based Chain. The transaction-based chain is a distributed ledger without blocks as a data structure. IOTA [64] is a blockchain project specially for IoT scenarios. Tangle [105] is the DAG blockchain of IOTA. It is directly composed of transactions without blocks, as shown in Figure 5. Transactions are appended to the Tangle without mining and transaction fees. In order to append a new transaction, a node has to choose two existing transactions from the Tangle and validate them. The two transactions may coincide. If the node approves these two transactions, two edges are generated to point to these two transactions from the new transaction. In this way, a new transaction is appended to Tangle with two pointers. Unlike Bitcoin, the transaction confirmation latency in IOTA is uncertain and unknown. Users can determine the transaction confirmation time by themselves. The more times a transaction is approved, the higher the reliability of the transaction. Similarly, Byteball [31] is also a DAG structured tx-based chain.

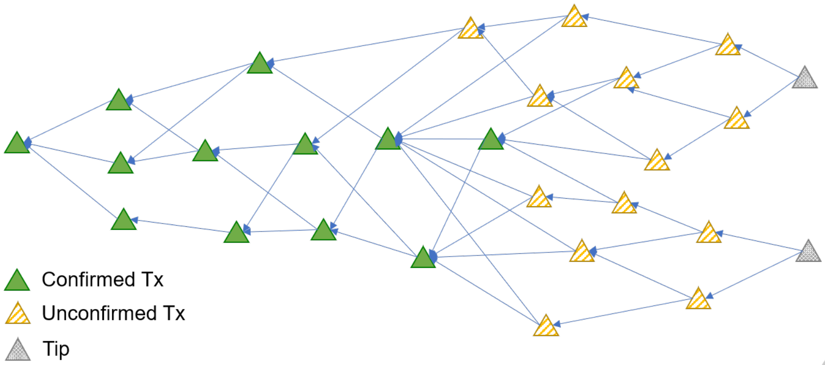

Fig. 5. The structure of Tangle: Each new transaction points to two previous transactions.

7.3.2 Hashgraph. Swirlds hashgraph consensus algorithm [11] is a completely asynchronous BFT protocol based on the DAG structure. It uniquely relies on *gossip about gossip* and *virtual vote* to reach consensus. There is no block, but a new data structure *event* in Hashgraph. Each event contains two pointers to its parent events. One parent event is its own last event, and the other is the event received from other nodes. Nodes sign their received events and continue to gossip to other nodes. Gossiping the history of the gossip is the *gossip about gossip* mechanism. It allows nodes to exchange and upgrade their local hashgraph. If the nodes all contain the same event, then the ancestors of the event and the corresponding edges are included. Because the event propagation indicates the node's vote for the event, each node can calculate the votes through its local hashgraph. This virtual vote is completely asynchronous and bandwidth-efficient.

7.3.3 Hyperledger Fabric. Hyperledger Fabric [4] is a permissioned blockchain, a sub-project of Hyperledger hosted by the Linux Foundation. Most BFT consensus protocols are based on order-execute architecture [4]: nodes first broadcast the sorted transactions, and then nodes execute the transactions sequentially. Uniquely, Fabric adopts the execute-order-validate architecture as follows.

- Execution Phase: The client sends a transaction proposal to specific nodes (called endorsers). The endorser simulates the execution of the transaction proposal on the specified smart contract (called chaincode) and outputs the result (writeset and readset). The chaincode is isolated and runs in a Docker container. The simulation is only on the local blockchain of the endorser. The endorser generates and sends the endorsement message to the client, which contains the signature of the simulation result and other information.
- Ordering Phase: The transaction is sent to the ordering service node if the client has collected enough endorsements for the transaction. The ordering service node packs transactions and generates a sequence of blocks. The generated block is broadcast to other peer nodes through gossip.
- Validation Phase: The validation system chaincode evaluates the endorsements of each transaction in a block in parallel. If it is invalid, the transaction will be discarded. Then, read-write conflict checks on the transaction in order. Finally, in the ledger update phase, peer nodes append the block to their local blockchain.

Fabric breaks through the performance bottleneck of executing all transactions sequentially by all peers in the blockchain using the order execution architecture [4]. Its modularity and versatility make it widely applicable to a variety of industrial scenarios.

7.3.4 Delegated Proof of Stake (DPoS) . DPoS does not use random encryption algorithms to select leaders and validators. Instead, all stakeholders can vote to determine who is trusted to generate blocks. In Bitshares [17], stakeholders vote to select nodes called witnesses at the beginning of each round. Witnesses act as representatives to generate, validate and broadcast blocks. Witnesses should pay deposits and can be rewarded after successfully generating blocks. Similarly, nodes in EOS [46] that get enough votes from stakeholders can become block generators. In each round, 21 nodes are selected to cooperate to generate new blocks. EOS combines asynchronous Byzantine Fault Tolerance (aBFT) to achieve faster transaction confirmation. In these DPoS protocols, the trust relationship needs to be established between nodes, which reduces the decentralization of the blockchain [81, 84].

7.3.5 Non-anonymous Proof-based Protocols. Ethereum Proof-of-Authority [21] is designed for enterprise-ready permissioned blockchain, and was released on Azure [7]. The system is not anonymous. Each participating node has one and only one known and reputable identity. All blocks are signed by their generators. All benefits nodes gained from making meaningful contributions are transparent, and so are malicious behaviors. The protocol adds new members, deletes compromised nodes, or elect administrators and validators by voting.

Gochain [58] is a blockchain in applying proof of reputation. Generally, reputation can be determined by influence, market value, and brand significance. The security of the proof of reputation blockchain depends on the reputation of participants. Only the node whose reputation has been validated can be regarded as the authoritative node. Only authoritative nodes can generate, sign new blocks and validate blocks. The list of authoritative nodes is recorded on the blockchain. If a node violates the rules, its reputation will be significantly affected.

7.3.6 Redactable Blockchain. Although immutability is an important feature of blockchains, it is not without flaws. Inappropriate or even illegal content is recorded on the blockchain, and unable to meet smart contract re-writable storage requirements, etc. Some blockchain consensus protocols [6, 39, 40] are committed to achieving redactable blockchain under the premise of ensuring transparency and auditability.

The literature [6] and [39] is two redactable blockchain schemes that use the chameleon hash instead of the traditional one-way hash to link blocks. The chameleon hash [76] is a special cryptographic function that contains a trap door. If there is no trap door key, the chameleon hash is considered collision-resistant, but once given the trap door key, collisions can be easily found. Thus, some nodes manage the trapdoor keys and can modify the block content without changing other blocks.

Deuber *et al.* [40] propose to implement a permissionless redactable blockchain by voting. Any node can initiate an edit request. The request contains the block index requested to be edited and the candidate block for replacement. After validating the edit request, miners vote the edit request by including the hash of the edit request in the blocks they mined. Once the edit request is approved, the candidate block is appended to the blockchain. Nodes validate the new block and check whether the subsequent votes meet the requirements before replacing the block.

8 COMPARISON AND DISCUSSION

In the previous sections, we introduced the mainstream proof-based consensus protocols and committee-based consensus protocols. We compare the characteristics of these two types of consensus protocols, as shown in Table 5.

These two types of consensus protocols represent two fundamental methodologies, reaching consensus by the proof of owning resources or by voting. The first is based on Nakamoto consensus protocol. It allows any node to participate in consensus protocols at any time, except for some special designs such as proof of authority and proof of reputation. The advantage is that there is no authority, no authorization, and it is fully decentralized. This setting also helps to achieve stronger anonymity. There is no connection between the account and the owner's identity. A user may have multiple accounts and these accounts are not censored. However, higher decentralization usually leads to lower throughput and longer confirmation latency. Another property of proof-based protocols is that they can only achieve probabilistic consensus. The consensus reached cannot represent

Table 5. Comparison of Proof-based & Committee-based consensus protocols

	Proof-based	**Committee-based**
Representative consensus algorithm	PoW, PoS	PBFT
Participation	Anytime	Conditional
Authorization	No	Yes
Decentralization	High	Low
Anonymity	Strong	Weak
Deterministic consensus	No	Most
Throughput	Low	High
Confirmation latency	Long	Short

the endorsement of participating nodes with a one hundred percent guarantee. Each round of consensus is probabilistic, but more rounds mean more endorsement by participating nodes with greater certainty. This is why the deeper the transaction, the lower the possibility of cancellation. More rounds instead of one round make the confirmation latency longer.

The committee-based protocols are derived from the classic distributed consensus protocols such as PBFT. A fixed number of participants form a committee and reach consensus. Before the start of the consensus protocols, the number of the participating nodes is known, and the identities of participating nodes are authenticated and transparent. In this case, anonymity is ignored because there is always an entity/mechanism to manage the joining of nodes. New nodes cannot dynamically participate in the consensus process. Generally, the committee-based consensus protocols can ensure that the consensus is agreed by participating nodes with the majority vote, and achieve deterministic consistency. Once a transaction is included in the block and added to the blockchain, the transaction is confirmed. Compared with proof-based consensus protocols, it can achieve higher throughput and shorter confirmation latency.

There is a trade-off between decentralization and throughput. Ideally, a consensus protocol should achieve both decentralization and high throughput. As shown in Figure 6, there are three evolutionary routes: the first is to retain the proof-based leader election method but adopt a new structure to achieve better concurrency instead of a single chain, such as DAG structure [82, 113], parallel-chains structure [10, 121, 129]. The second is to expand the committee-based protocols to allow more nodes to join. One representative approach is sharding [34, 75, 87, 130]. Nodes are randomly selected and divided into multiple shards to participate in the consensus process in parallel. The third is the hybrid model, which combines the features of the proof-based and committee-based schemes. For example, in BFT-based PoS consensus protocols [24, 57, 106], there is no authority to validate the identity of participating nodes, as long as the nodes can prove themselves have valid stakes. Some nodes are selected to join the committee. It is more efficient for a limited number of committee members to reach consensus, and the blockchain throughput is also improved.

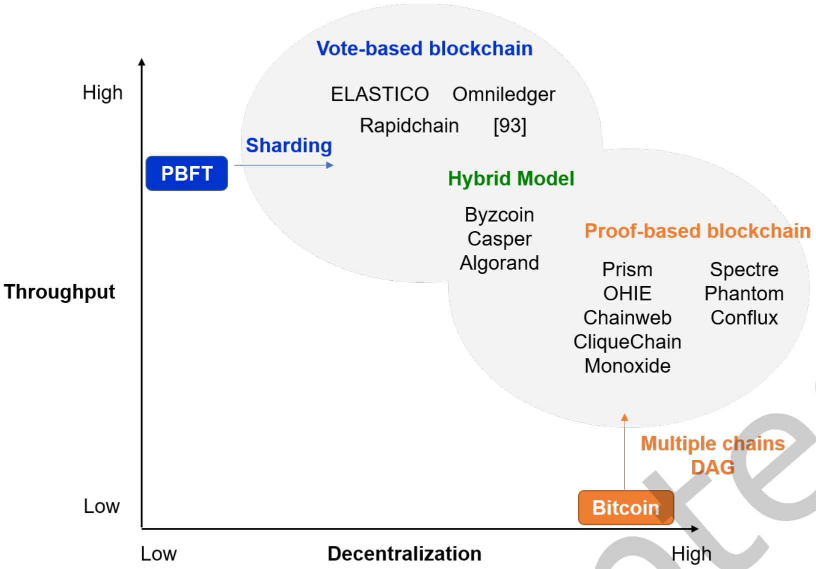

Fig. 6. Improvements of blockchain consensus protocols

Table 6. Comparison of PoW-based and PoS-based consensus protocols

	PoW	PoS
Weight	Computing power	Cryptocurrency
Application	Permissionless	Permissionless
Energy consumption	Massive	Less
Fork	High frequency	Less
Attacks	Double spending attack	Nothing at stake attack
	Self-mining attack	Stake grinding attack
	Eclipse attack	Stake bleeding attack

PoW and PoS are most commonly used in proof-based consensus protocols. Their main features are compared as shown in Table 6. PoS-based consensus protocols no longer require computing power consumption. Nodes holding some cryptocurrency/stakes can participate in the consensus process, which is more friendly to nodes with low computing power. However, it is arguable that the decentralization of PoS-based blockchain is weaker than the PoW-based blockchain. A small number of nodes own most of the cryptocurrency in a PoS-based blockchain system. Nodes with more cryptocurrencies in the PoS-based consensus protocol are more likely to generate new blocks. It is easy to make the rich get richer, resulting in centralization and security issues. The stakeholders may mortgage or delegate the stakes to other nodes for profit. It further weakens decentralization.

According to the assumptions of network models, committee-based blockchain consensus protocols can be divided into three categories: synchronous, partially synchronous, and asynchronous. The consensus protocols based on synchronous settings can tolerate up to $n/2$ Byzantine nodes, but protocols based on partially synchronous or asynchronous settings can tolerate up to $n/3$ Byzantine nodes [94]. Synchronous or partially synchronous network models assume that there is a time-bound for message propagation. Synchronous or partially synchronous network models are suitable for well-connected permissioned blockchains. The asynchronous network model is more in line with more complex underlying networks, but deterministic protocols cannot be realized because of FLP impossibility.

9 POTENTIAL FUTURE RESEARCH

Despite extensive research into blockchain consensus protocols, the performance of consensus is still not satisfactory. In this section, we list some remaining unsolved challenges and propose potential research directions for blockchain consensus.

9.1 Efficient Transactions Processing

One important research direction is to minimize duplicate/conflicting transactions on the blockchain while allowing concurrent blocks to be appended to the blockchain. Regardless of DAG blockchains, parallel blockchains, or sharding-based blockchain, there are transaction conflicts and duplication when multiple blocks are allowed to be appended to the blockchain concurrently. Most blockchain scaling schemes aim to achieve higher throughput, but they ignore the effectiveness of transactions recorded on the blockchain. Adding conflicting or duplicate transactions on the blockchain hinders the effective throughput of the blockchain. We can use transaction information to allocate transactions to reduce duplicate transactions between concurrent blocks, such as the distance between miners' identities and transactions' identities. It is also possible to reduce conflicts by prioritizing transactions based on the transactions' identities or miners' identities. If there are no duplicate or conflicting transactions on the blockchain, the effective throughput of the system will be higher.

Another research direction is to improve the efficiency of cross-shard transaction processing. A common method for sharding-based blockchains is to randomly divide transactions into shards for load balancing. The cross-shard communication is required to validate transactions when the inputs of a transaction are in different shards. It adds high communication costs. The cross-shard transaction processing is the performance bottleneck of sharding-based blockchains. Further research should be conducted on efficient cross-shard transaction processing for higher throughput and lower confirmation latency.

9.2 Performance Improvement of Committee-based BFT Protocols

Committee-based BFT protocols can achieve high throughput with low computing costs. But they are communication-intensive protocols that are only efficient when applied to a limited number of nodes. Communication costs limit scalability and decentralization when they are applied to large-scale networks.

A research direction is to use cryptographic algorithms, such as threshold signatures, collective signatures, etc., to reduce the communication costs of committee-based BFT protocols. In this case, it is no longer necessary for each node to receive/send messages with the signature from/to all other nodes. Avoiding all-to-all gossip can reduce the communication complexity to $O(n)$ or even less. Reducing communication costs of committee-based BFT protocols helps increase the number of nodes participating in consensus and improves the decentralization of blockchains.

Another direction is to improve committee-based BFT protocols when they are extended to permissionless environments. Committee-based BFT protocols are closed. In each round, only a part of nodes is selected to participate in the consensus with communication, computing, and storage resources. However, when they are extended to BFT-based permissionless blockchains, only selected nodes can participate in the consensus, and other nodes cannot participate in the consensus even if they have abundant resources. Massive resources are idle. If these idle resources are used more efficiently through methods such as decoupling functions and asynchronous processing, the performance of BFT-based permissionless blockchains can be further improved.

9.3 Cross-chain Interoperability

Blockchains are expected to become interconnected networks in the future. A single blockchain can only achieve limited functions. Different blockchains have different participants, run different consensus protocols, and can execute different functions. In addition to the performance improvement of a single blockchain, research on the performance improvement of the collaboration of multiple blockchains is also significant. Interoperability may become an important criterion when measuring blockchains.

One potential direction is sidechains. Sidechains allow multiple blockchains to communicate with each other and perform various cross-chain operations through the two-way peg. A blockchain can be a sidechain of another blockchain. Data or assets can be exchanged safely and efficiently between different blockchains. Although sidechains enable two-way communication between blockchains, they can maintain security isolation. The attacks on one chain cannot compromise the security of another chain.

Besides, we can adopt a heterogeneous architecture to achieve the interoperability of chains in different shards. For example, in addition to multiple chains in shards, there is a Beacon chain for interoperability in Ethereum 2.0. As the core component of Ethereum 2.0, Beacon chain is responsible for coordinating chains in different shards. Beacon chain and the chains in shards are tightly coupled. Beacon chain frequently communicates with blockchains in other shards to maintain the consistency of the entire system.

10 CONCLUSION

In this survey, we summarized the latest development of blockchain consensus protocols. Generally, proof-based blockchain consensus protocols inherit the design of the Nakamoto consensus protocol, and they have shortcomings, such as low throughput and long confirmation latency; while committee-based consensus protocols follow the design of classical consensus protocols for distributed systems, but are not suitable for large-scale networks. We also discussed the unsolved challenges in the design of blockchain consensus protocols and the future research directions.

ACKNOWLEDGMENTS

This work is supported by the Hong Kong RGC GRF under the Grant No. CityU 11213920 and RGC RIF No. R1012-21.

REFERENCES

[1] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2017. Solida: A Blockchain Protocol Based on Reconfigurable Byzantine Consensus. Cryptology ePrint Archive, Report 2017/1118. (2017). https://ia.cr/2017/1118.

[2] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Maofan Yin. 2020. Sync hotstuff: Simple and practical synchronous state machine replication. In *Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 106–118. https://doi.org/10.1109/SP40000.2020.00044

[3] Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. 2019. Asymptotically Optimal Validated Asynchronous Byzantine Agreement. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC '19)*. Association for Computing Machinery, New York, NY, USA, 337–346. https://doi.org/10.1145/3293611.3331612

[4] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference (EuroSys '18)*. Association for Computing Machinery, New York, NY, USA, Article 30, 15 pages. https://doi.org/10.1145/3190508.3190538

[5] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. 2007. Provable Data Possession at Untrusted Stores. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*. Association for Computing Machinery, New York, NY, USA, 598–609. https://doi.org/10.1145/1315245.1315318

[6] Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton Andrade. 2017. Redactable Blockchain - or - Rewriting History in Bitcoin and Friends. In *Proceedings of the 2017 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, Los Alamitos, CA,

USA, 111–126. https://doi.org/10.1109/EuroSP.2017.37

[7] Azure. Azure . (????). Retrieved November 3, 2021 from https://azure.microsoft.com/en-us/

[8] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. 2014. Enabling blockchain innovations with pegged sidechains. (2014). http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains

[9] Christian Badertscher, Peter Gaži, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. 2018. Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. Association for Computing Machinery, New York, NY, USA, 913–930. https://doi.org/10.1145/3243734.3243848

[10] Vivek Bagaria, Sreeram Kannan, David Tse, Giulia Fanti, and Pramod Viswanath. 2019. Prism: Deconstructing the Blockchain to Approach Physical Limits. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. Association for Computing Machinery, New York, NY, USA, 585–602. https://doi.org/10.1145/3319535.3363213

[11] Leemon Baird. 2016. The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance. (May 2016). https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf

[12] Michael Ben-Or, Boaz Kelmer, and Tal Rabin. 1994. Asynchronous Secure Computations with Optimal Resilience. In *Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing (PODC '94)*. Association for Computing Machinery, New York, NY, USA, 183–192. https://doi.org/10.1145/197917.198088

[13] Juan Benet and Nicola Greco. 2017. Filecoin: A decentralized storage network. (2017). https://filecoin.io/filecoin.pdf

[14] Iddo Bentov, Rafael Pass, and Elaine Shi. 2016. Snow White: Provably Secure Proofs of Stake. (2016). https://allquantor.at/blockchainbib/pdf/bentov2016snow.pdf

[15] Alex Biryukov and Dmitry Khovratovich. 2016. Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem. In *Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS'16)*. The Internet Society, Rosten, VA, USA. http://dx.doi.org/10.14722/ndss.2016.23108

[16] George Bissias and Brian N Levine. 2020. Bobtail: improved blockchain security with low-variance mining. In *Proceedings of the 27th Annual Network and Distributed System Security Symposium (NDSS'20)*. The Internet Society, Rosten, VA, USA, 1–16. https://dx.doi.org/10.14722/ndss.2020.23095

[17] Bitshares. Bitshares. (????). Retrieved November 3, 2021 from https://bitshares.org/

[18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. 2018. Verifiable delay functions. In *Proceedings of the 38th Annual International Cryptology Conference*, Vol. 10991. Springer, 757–788. https://doi.org/10.1007/978-3-319-96884-1_25

[19] Dan Boneh, Benedikt Bünz, and Ben Fisch. 2018. A Survey of Two Verifiable Delay Functions. Cryptology ePrint Archive, Report 2018/712. (2018). https://ia.cr/2018/712.

[20] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. 2015. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 104–121. https://doi.org/10.1109/SP.2015.14

[21] Cody Born. 2018. Ethereum Proof-of-Authority on Azure. (August 2018). https://azure.microsoft.com/en-us/blog/ethereum-proof-of-authority-on-azure/

[22] Kevin D. Bowers, Ari Juels, and Alina Oprea. 2009. Proofs of Retrievability: Theory and Implementation. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security (CCSW '09)*. Association for Computing Machinery, New York, NY, USA, 43–54. https://doi.org/10.1145/1655008.1655015

[23] Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. (2014). https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[24] Vitalik Buterin and Virgil Griffith. 2019. Casper the Friendly Finality Gadget. (2019). arXiv:cs.CR/1710.09437

[25] Christian Cachin and Marko Vukolić. 2017. Blockchain Consensus Protocols in the Wild. (2017). arXiv:cs.DC/1707.01873

[26] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine fault tolerance. In *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI' 99)*. USENIX Association, New Orleans, LA, 173–186.

[27] Miguel Castro and Barbara Liskov. 2002. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)* 20, 4 (2002), 398–461.

[28] T-H. Hubert Chan, Rafael Pass, and Elaine Shi. 2018. PaLa: A Simple Partially Synchronous Blockchain. Cryptology ePrint Archive, Report 2018/981. (2018). https://ia.cr/2018/981.

[29] T-H. Hubert Chan, Rafael Pass, and Elaine Shi. 2018. PiLi: An Extremely Simple Synchronous Blockchain. Cryptology ePrint Archive, Report 2018/980. (2018). https://ia.cr/2018/980.

[30] Chia. Chia. (????). Retrieved November 3, 2021 from https://www.chia.net/

[31] Anton Churyumov. 2018. Byteball: A Decentralized System for Storage and Transfer of Value. (2018). https://byteball.org/Byteball.pdf

[32] Nxt community. 2016. Nxt Whitepaper. (2016). https://nxtdocs.jelurida.com/Nxt_Whitepaper

[33] Matt Corallo. 2016. Compact Block Relay. (2016). https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki

[34] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. 2019. Towards Scaling Blockchain Systems via Sharding. In *Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 123–140. https://doi.org/10.1145/3299869.3319889

[35] DASH. DASH. (????). Retrieved November 3, 2021 from https://www.dash.org/

[36] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. 2018. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *Proceedings of the 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vol. 10821. Springer, 66–98. https://doi.org/10.1007/978-3-319-78375-8_3

[37] Soubhik Deb, Sreeram Kannan, and David Tse. 2021. PoSAT: proof-of-work availability and unpredictability, without the work. In *Proceedings of the 25th International Conference on Financial Cryptography and Data Security*, Vol. 12675. Springer, Berlin, Heidelberg, 104–128. https://doi.org/10.1007/978-3-662-64331-0_6

[38] Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. 2020. Everything is a Race and Nakamoto Always Wins. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*. Association for Computing Machinery, New York, NY, USA, 859–878. https://doi.org/10.1145/3372297.3417290

[39] David Derler, Kai Samelin, Daniel Slamanig, and Christoph Striecks. 2019. Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based. In *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS'19)*. The Internet Society, Rosten, VA, USA, 1–15. https://dx.doi.org/10.14722/ndss.2019.23066

[40] Dominic Deuber, Bernardo Magri, and Sri Aravinda Krishnan Thyagarajan. 2019. Redactable blockchain in the permissionless setting. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 124–138. https://doi.org/10.1109/SP.2019.00039

[41] Dogecoin. Dogecoin. (????). Retrieved November 3, 2021 from https://dogecoin.com/

[42] Sisi Duan, Michael K. Reiter, and Haibin Zhang. 2018. BEAT: Asynchronous BFT Made Practical. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. Association for Computing Machinery, New York, NY, USA, 2028–2041. https://doi.org/10.1145/3243734.3243812

[43] Evan Duffield and Daniel Diaz. 2018. Dash: A Payments-Focused Cryptocurrency. (August 2018). https://github.com/dashpay/dash/wiki/Whitepaper

[44] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. 1988. Consensus in the Presence of Partial Synchrony. *J. ACM* 35, 2 (April 1988), 288–323. https://doi.org/10.1145/42282.42283

[45] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2013. Proofs of Space. Cryptology ePrint Archive, Report 2013/796. (2013). https://ia.cr/2013/796.

[46] EOS. EOS. (????). Retrieved November 3, 2021 from https://eos.io/

[47] Ethereum. Ethereum. (????). Retrieved November 3, 2021 from https://ethereum.org/en/

[48] Ittay Eyal. 2015. The Miner's Dilemma. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 89–103. https://doi.org/10.1109/SP.2015.13

[49] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol. In *Proceedings of the 13th USENIX symposium on networked systems design and implementation (NSDI' 16)*. USENIX Association, Santa Clara, CA, USA, 45–59. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal

[50] Ittay Eyal and Emin Gün Sirer. 2018. Majority is Not Enough: Bitcoin Mining is Vulnerable. *Commun. ACM* 61, 7, 95–102. https://doi.org/10.1145/3212998

[51] filecoin. Filecoin Documentation. (????). Retrieved November 3, 2021 from https://docs.filecoin.io/

[52] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of Distributed Consensus with One Faulty Process. *J. ACM* 32, 2 (April 1985), 374–382. https://doi.org/10.1145/3149.214121

[53] Chaya Ganesh, Claudio Orlandi, and Daniel Tschudi. 2019. Proof-of-Stake Protocols for Privacy-Aware Blockchains. In *Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer International Publishing, 690–719.

[54] Yingzi Gao, Yuan Lu, Zhenliang Lu, Qiang Tang, Jing Xu, and Zhenfeng Zhang. 2022. Dumbo-NG: Fast Asynchronous BFT Consensus with Throughput-Oblivious Latency. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1187–1201. https://doi.org/10.1145/3548606.3559379

[55] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The bitcoin backbone protocol: Analysis and applications. In *Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vol. 9057. Springer, 281–310. https://doi.org/10.1007/978-3-662-46803-6_10

[56] Peter Gaži, Aggelos Kiayias, and Dionysis Zindros. 2019. Proof-of-stake sidechains. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 139–156. https://doi.org/10.1109/SP.2019.00040

[57] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17)*. Association for Computing Machinery, New York, NY, USA, 51–68. https://doi.org/10.1145/3132747.3132757

[58] Gochain. Gochain. (????). Retrieved November 3, 2021 from https://gochain.io/

[59] Bingyong Guo, Yuan Lu, Zhenliang Lu, Qiang Tang, Jing Xu, and Zhenfeng Zhang. 2022. Speeding Dumbo: Pushing Asynchronous BFT Closer to Practice. In *Proceedings of the 29th Network and Distributed System Security Symposium (NDSS'22)*. The Internet Society, Renton, WA, USA, 1–18. https://www.ndss-symposium.org/wp-content/uploads/2022-385-paper.pdf

[60] Bingyong Guo, Zhenliang Lu, Qiang Tang, Jing Xu, and Zhenfeng Zhang. 2020. Dumbo: Faster Asynchronous BFT Protocols. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*. Association for Computing Machinery, New York, NY, USA, 803–818. https://doi.org/10.1145/3372297.3417262

[61] Zicong Hong, Song Guo, Peng Li, and Wuhui Chen. 2021. Pyramid: A layered sharding blockchain system. In *Proceedings of the 2021 IEEE Conference on Computer Communications*. IEEE, 1–10.

[62] Huawei Huang, Xiaowen Peng, Jianzhou Zhan, Shenyang Zhang, Yue Lin, Zibin Zheng, and Song Guo. 2022. BrokerChain: A Cross-Shard Blockchain Protocol for Account/Balance-based State Sharding. In *Proceedings of the 2022 IEEE Conference on Computer Communications*.

[63] Jun Huang, Debiao He, Mohammad S. Obaidat, Pandi Vijayakumar, Min Luo, and Kim-Kwang Raymond Choo. 2021. The Application of the Blockchain Technology in Voting Systems: A Review. *ACM Comput. Surv.* 54, 3, Article 60 (April 2021), 28 pages. https://doi.org/10.1145/3439725

[64] Iota. IOTA. (????). Retrieved November 3, 2021 from https://www.iota.org/

[65] Ari Juels and Burton S. Kaliski. 2007. Pors: Proofs of Retrievability for Large Files. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*. Association for Computing Machinery, New York, NY, USA, 584–597. https://doi.org/10.1145/1315245.1315317

[66] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20, 1 (1998), 359–392.

[67] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. 2020. Order-fairness for byzantine consensus. In *Proceedings of the 40th Annual International Cryptology Conference*, Vol. 12172. Springer, 451–480. https://doi.org/10.1007/978-3-030-56877-1_16

[68] Thomas Kerber, Aggelos Kiayias, Markulf Kohlweiss, and Vassilis Zikas. 2019. Ouroboros Crypsinous: Privacy-Preserving Proof-of-Stake. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 157–174. https://doi.org/10.1109/SP.2019.00063

[69] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Proceedings of the 37th Annual International Cryptology Conference*, Vol. 10401. Springer, 357–388. https://doi.org/10.1007/978-3-319-63688-7_12

[70] Lucianna Kiffer, Rajmohan Rajaraman, and abhi shelat. 2018. A Better Method to Analyze Blockchain Consistency. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. Association for Computing Machinery, New York, NY, USA, 729–744. https://doi.org/10.1145/3243734.3243814

[71] Sunny King. 2013. Primecoin: Cryptocurrency with prime number proof-of-work. (July 2013). Retrieved October 29, 2021 from https://primecoin.io/bin/primecoin-paper.pdf

[72] Sunny King and Scott Nadal. 2012. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. (August 2012). Retrieved October 29, 2021 from https://www.peercoin.net/whitepapers/peercoin-paper.pdf

[73] Markulf Kohlweiss, Varun Madathil, Kartik Nayak, and Alessandra Scafuro. 2021. On the Anonymity Guarantees of Anonymous Proof-of-Stake Protocols. In *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 1818–1833. https://doi.org/10.1109/SP40001.2021.00107

[74] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. 2016. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium (USENIX Security' 16)*. USENIX Association, Austin, TX, 279–296.

[75] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. 2018. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 583–598. https://doi.org/10.1109/SP.2018.000-5

[76] Hugo Krawczyk and Tal Rabin. 2000. Chameleon signatures. In *Proceedings of the Network and Distributed System Security Symposium(NDSS'00)*. The Internet Society, Rosten, VA, USA, 1–12. https://www.ndss-symposium.org/ndss2000/chameleon-signatures/

[77] Jae Kwon. 2014. Tendermint: Consensus without mining. (2014). https://tendermint.com/static/docs/tendermint.pdf

[78] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 2017. Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. Association for Computing Machinery, New York, NY, USA, 195–209. https://doi.org/10.1145/3133956.3134019

[79] Leslie Lamport. 1998. The Part-Time Parliament. *ACM Trans. Comput. Syst.* 16, 2 (May 1998), 133–169. https://doi.org/10.1145/279227.279229

[80] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4, 3 (July 1982), 382–401. https://doi.org/10.1145/357172.357176

[81] Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. 2020. A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus, and Traffic Modeling. *ACM Comput. Surv.* 53, 1, Article 18 (Feb. 2020), 32 pages. https://doi.org/10.1145/3372136

[82] Chenxing Li, Peilun Li, Dong Zhou, Wei Xu, Fan Long, and Andrew Yao. 2018. Scaling Nakamoto Consensus to Thousands of Transactions per Second. (2018). arXiv:cs.DC/1805.03870

[83] Chenxing Li, Peilun Li, Dong Zhou, Zhe Yang, Ming Wu, Guang Yang, Wei Xu, Fan Long, and Andrew Chi-Chih Yao. 2020. A Decentralized Blockchain with High Throughput and Fast Confirmation. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association, 515–528. https://www.usenix.org/conference/atc20/presentation/li-chenxing

[84] Xi Li, Zehua Wang, Victor C. M. Leung, Hong Ji, Yiming Liu, and Heli Zhang. 2021. Blockchain-Empowered Data-Driven Networks: A Survey and Outlook. *ACM Comput. Surv.* 54, 3, Article 58 (apr 2021), 38 pages. https://doi.org/10.1145/3446373

[85] Xing Li, Yi Zheng, Kunxian Xia, Tongcheng Sun, and John Beyler. 2020. Phantom: An Efficient Privacy Protocol Using zk-SNARKs Based on Smart Contracts. Cryptology ePrint Archive, Report 2020/156. (2020). https://ia.cr/2020/156.

[86] Litecoin. Litecoin. (????). Retrieved November 3, 2021 from https://litecoin.org/

[87] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. 2016. A Secure Sharding Protocol For Open Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 17–30. https://doi.org/10.1145/2976749.2978389

[88] Lynx. Lynx. (????). Retrieved November 3, 2021 from https://getlynx.io/

[89] Dahlia Malkhi, Kartik Nayak, and Ling Ren. 2019. Flexible Byzantine Fault Tolerance. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. Association for Computing Machinery, New York, NY, USA, 1041–1053. https://doi.org/10.1145/3319535.3354225

[90] Will Martino, Monica Quaintance, and Stuart Popejoy. 2018. Chainweb: A proof-of-work parallel-chain architecture for massive throughput. (2018). https://d31d887a-c1e0-47c2-aa51-c69f9f998b07.filesusr.com/ugd/86a16f_029c9991469e4565a7c334dd716345f4.pdf

[91] Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. 2014. Permacoin: Repurposing Bitcoin Work for Data Preservation. In *Proceedings of 2014 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 475–490. https://doi.org/10.1109/SP.2014.37

[92] Andrew Miller, Ahmed Kosba, Jonathan Katz, and Elaine Shi. 2015. Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. Association for Computing Machinery, New York, NY, USA, 680–691. https://doi.org/10.1145/2810103.2813621

[93] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The Honey Badger of BFT Protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 31–42. https://doi.org/10.1145/2976749.2978399

[94] Atsuki Momose and Ling Ren. 2021. Multi-Threshold Byzantine Fault Tolerance. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*. Association for Computing Machinery, New York, NY, USA, 1686–1699. https://doi.org/10.1145/3460120.3484554

[95] Monero. Monero. (????). Retrieved November 3, 2021 from https://www.getmonero.org/

[96] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer misc cash system. (2008). http://www.bitcoin.org/bitcoin.pdf

[97] Lan N Nguyen, Truc DT Nguyen, Thang N Dinh, and My T Thai. 2019. Optchain: optimal transactions placement for scalable blockchain sharding. In *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 525–535.

[98] Brian M. Oki and Barbara H. Liskov. 1988. Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems. In *Proceedings of the Seventh Annual ACM Symposium on Principles of Distributed Computing (PODC '88)*. Association for Computing Machinery, New York, NY, USA, 8–17. https://doi.org/10.1145/62546.62549

[99] Diego Ongaro and John Ousterhout. 2014. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC '14)*. USENIX Association, Philadelphia, PA, USA, 305–319. https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro

[100] Sunoo Park, Albert Kwon, Georg Fuchsbauer, Peter Gaži, Joël Alwen, and Krzysztof Pietrzak. 2018. SpaceMint: A Cryptocurrency Based on Proofs of Space. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security*, Vol. 10957. Springer, Berlin, Heidelberg, 480–499. https://doi.org/10.1007/978-3-662-58387-6_26

[101] Rafael Pass, Lior Seeman, and Abhi Shelat. 2017. Analysis of the blockchain protocol in asynchronous networks. In *Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vol. 10211. Springer, 643–673. https://doi.org/10.1007/978-3-319-56614-6_22

[102] Rafael Pass and Elaine Shi. 2017. FruitChains: A Fair Blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '17)*. Association for Computing Machinery, New York, NY, USA, 315–324. https://doi.org/10.1145/3087801.3087809

[103] Rafael Pass and Elaine Shi. 2017. The Sleepy Model of Consensus. In *Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security*. Springer, 380–409. https://doi.org/10.1007/978-3-319-70697-9_14

[104] Colin Percival and Simon Josefsson. 2016. The scrypt password-based key derivation function. *IETF Draft URL: http://tools. ietf. org/html/josefsson-scrypt-kdf-00. txt (accessed: 30.11. 2012)* (2016).

[105] Serguei Popov. 2018. The Tangle. (April 2018). http://www.descryptions.com/Iota.pdf

[106] Daniël Reijsbergen, Pawel Szalachowski, Junming Ke, Zengpeng Li, and Jianying Zhou. 2021. ProPoS: A Probabilistic Proof-of-Stake Protocol. In *Proceedings of the 28th Network and Distributed System Security Symposium (NDSS'21)*. The Internet Society, Rosten, VA, USA, 1–18. https://dx.doi.org/10.14722/ndss.2021.24164

[107] Ling Ren and Srinivas Devadas. 2016. Proof of space from stacked expanders. In *Proceedings of the 14th International Conference on Theory of Cryptography*, Vol. 9985. Springer, Berlin, Heidelberg, 262–285. https://doi.org/10.1007/978-3-662-53641-4_11

[108] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from bitcoin. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 459–474. https://doi.org/10.1109/SP.2014.36

[109] Sawtooth. Hyperledger Sawtooth. (????). Retrieved November 3, 2021 from https://www.hyperledger.org/use/sawtooth

[110] Peiyao Sheng, Gerui Wang, Kartik Nayak, Sreeram Kannan, and Pramod Viswanath. 2021. BFT Protocol Forensics. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*. Association for Computing Machinery, New York, NY, USA, 1722–1743. https://doi.org/10.1145/3460120.3484566

[111] Ali Shoker. 2018. Brief Announcement: Sustainable Blockchains through Proof of EXercise. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC '18)*. Association for Computing Machinery, New York, NY, USA, 269–271. https://doi.org/10.1145/3212734.3212781

[112] Amritraj Singh, Kelly Click, Reza M. Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2020. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications* 149 (2020), 102471. https://doi.org/10.1016/j.jnca.2019.102471

[113] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. 2016. SPECTRE: A Fast and Scalable Cryptocurrency Protocol. Cryptology ePrint Archive, Report 2016/1159. (2016). https://ia.cr/2016/1159.

[114] Yonatan Sompolinsky and Aviv Zohar. 2015. Secure high-rate transaction processing in bitcoin. In *Proceedings of the 19th International Conference on Financial Cryptography and Data Security*, Vol. 8975. Springer, Berlin, Heidelberg, 507–527. https://doi.org/10.1007/978-3-662-47854-7_32

[115] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J Fischer, and Bryan Ford. 2017. Scalable bias-resistant distributed randomness. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 444–460. https://doi.org/10.1109/SP.2017.45

[116] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. 2016. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 526–545. https://doi.org/10.1109/SP.2016.38

[117] Pawel Szalachowski, Daniël Reijsbergen, Ivan Homoliak, and Siwei Sun. 2019. Strongchain: Transparent and collaborative proof-of-work consensus. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security' 19)*. USENIX Association, Santa Clara, CA, USA, 819–836. https://www.usenix.org/conference/usenixsecurity19/presentation/szalachowski

[118] Yuechen Tao, Bo Li, Jingjie Jiang, Hok Chu Ng, Cong Wang, and Baochun Li. 2020. On sharding open blockchains with smart contracts. In *Proceedings of the 36th International Conference on Data Engineering (ICDE)*. IEEE, 1357–1368.

[119] Tendermint. Tendermint. (????). Retrieved November 3, 2021 from https://tendermint.com/

[120] Huibo Wang, Guoxing Chen, Yinqian Zhang, and Zhiqiang Lin. 2022. Multi-Certificate Attacks Against Proof-of-Elapsed-Time And Their Countermeasures. In *Proceedings of the 29th Network and Distributed System Security Symposium (NDSS'22)*. The Internet Society, Rosten, VA, USA, 1–17. https://www.ndss-symposium.org/wp-content/uploads/2022-158-paper.pdf

[121] Jiaping Wang and Hao Wang. 2019. Monoxide: Scale out blockchains with asynchronous consensus zones. In *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI' 19)*. USENIX Association, Boston, MA, 95–112. https://www.usenix.org/conference/nsdi19/presentation/wang-jiaping

[122] Xuechao Wang, Viswa Virinchi Muppirala, Lei Yang, Sreeram Kannan, and Pramod Viswanath. 2021. Securing Parallel-chain Protocols under Variable Mining Power. (2021). arXiv:cs.CR/2105.02927

[123] Gavin Wood. 2021. Ethereum: A secure decentralised generalised transaction ledger. (November 2021). https://ethereum.github.io/yellowpaper/paper.pdf

[124] Cheng Xu, Ce Zhang, Jianliang Xu, and Jian Pei. 2021. SlimChain: scaling blockchain transactions through off-chain storage and parallel processing. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2314–2326.

[125] Jie Xu, Yingying Cheng, Cong Wang, and Xiaohua Jia. 2021. Occam: A Secure and Adaptive Scaling Protocol for Permissionless Blockchain. In *Proceedings of the 41st IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Los Alamitos, CA, USA, 618–628. https://doi.org/10.1109/ICDCS51616.2021.00065

[126] Lei Yang, Vivek Bagaria, Gerui Wang, Mohammad Alizadeh, David Tse, Giulia Fanti, and Pramod Viswanath. 2020. Prism: Scaling Bitcoin by 10,000x. (2020). arXiv:cs.DC/1909.11261

[127] Lei Yang, Seo Jin Park, Mohammad Alizadeh, Sreeram Kannan, and David Tse. 2022. DispersedLedger: High-Throughput Byzantine Consensus on Variable Bandwidth Networks. In *Proceedings of the 19th USENIX symposium on networked systems design and implementation*

(NSDI' 19). USENIX Association, Santa Clara, CA, USA, 493–512. https://www.usenix.org/system/files/nsdi22-paper-yang_lei.pdf

[128] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus with Linearity and Responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC '19)*. Association for Computing Machinery, New York, NY, USA, 347–356. https://doi.org/10.1145/3293611.3331591

[129] Haifeng Yu, Ivica Nikolić, Ruomu Hou, and Prateek Saxena. 2020. OHIE: Blockchain scaling made simple. In *Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 90–105. https://doi.org/10.1109/SP40000.2020.00008

[130] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. 2018. RapidChain: Scaling Blockchain via Full Sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. Association for Computing Machinery, New York, NY, USA, 931–948. https://doi.org/10.1145/3243734.3243853

[131] Zcash. Zcash. (????). Retrieved November 3, 2021 from https://z.cash/

[132] Ren Zhang and Bart Preneel. 2019. Lay Down the Common Metrics: Evaluating Proof-of-Work Consensus Protocols' Security. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, Los Alamitos, CA, USA, 175–192. https://doi.org/10.1109/SP.2019.00086

[133] Ren Zhang, Dingwei Zhang, Quake Wang, Shichen Wu, Jan Xie, and Bart Preneel. 2022. NC-Max: Breaking the Security-Performance Tradeoff in Nakamoto Consensus. In *Proceedings of the 29th Network and Distributed System Security Symposium (NDSS'22)*. The Internet Society, Rosten, VA, USA, 1–18. https://www.ndss-symposium.org/wp-content/uploads/2022-370-paper.pdf

[134] Yunhao Zhang, Srinath Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. 2020. Byzantine Ordered Consensus without Byzantine Oligarchy. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI' 20)*. USENIX Association, 633–649. https://www.usenix.org/conference/osdi20/presentation/zhang-yunhao

Received 24 November 2021; revised 22 November 2022; accepted 3 January 2023