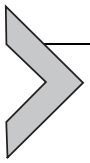


CHAPTER SEVEN

Consensus mechanisms and information security technologies

Peng Zhang[a], **Douglas C. Schmidt**[b], **Jules White**[b], **Abhishek Dubey**[b]
[a]Department of Biomedical Informatics, Vanderbilt University Medical Center, Nashville, TN, United States
[b]Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, United States

Contents

Abstract

Distributed Ledger Technology (DLT) helps maintain and distribute predefined types of information and data in a decentralized manner. It removes the reliance on a third-party intermediary, while securing information exchange and creating shared truth via transaction records that are hard to tamper with. The successful operation of DLT stems

Advances in Computers, Volume 115
ISSN 0065-2458
https://doi.org/10.1016/bs.adcom.2019.05.001

largely from two computer science technologies: consensus mechanisms and information security protocols. Consensus mechanisms, such as Proof of Work (PoW) and Raft, ensure that the DLT network collectively agrees on contents stored in the ledger. Information security protocols, such as encryption and hashing, protect data integrity and safeguard data against unauthorized access.

The most popular incarnation of DLT has been used in cryptocurrencies, such as Bitcoin and Ethereum, through public blockchains, which requires the application of more robust consensus protocols across the entire network. An example is PoW, which has been employed by Bitcoin, but which is also highly energy inefficient. Other forms of DLT include consortium and private blockchains where networks are configured within federated entities or a single organization, in which case less energy intensive consensus protocols (such as Raft) would suffice. This chapter surveys existing consensus mechanisms and information security technologies used in DLT.

1. Introduction

Blockchain technologies alleviate the reliance on a centralized authority to certify information integrity and ownership, as well as mediate transactions and exchange of digital assets, while enabling secure and pseudo-anonymous transactions along with agreements directly between interacting parties. Since the introduction of Bitcoin by Satoshi Nakamoto [1], blockchain technology has been studied by researchers, engineers, and domain experts to evaluate its utility and improve its usability. Bitcoin is the first successful application of blockchain technology that is widely recognized for its revolutionary mechanisms that allow the secure direct transfer of digital assets between involved parties without the need for a trusted intermediary.

The concept and importance of digital assets are integral to the inception of blockchain technology. A digital asset is anything that exists in a binary format that comes with some right to exercise. One type of digital assets is native assets, which are assets that lack physical substance, but can be owned or controlled to produce some value. Examples of native assets are digital music, images, movies, electronic funds, and software. The other type of digital assets is digital representations of traditional assets, which are or historically used to exist in paper certificates or titles [2]. Assets like land property, gold, automobile title, and "paper" currency are examples of this type of digital assets.

The global economy increasingly depends on the effective management of digital assets [3] in nearly every domain and aspect of our lives. For example, the entertainment industry requires digital rights management for

movies and music; the finance industry has experienced far more electronic fund transfers than cash exchanges; the energy sector is moving toward digital trading of energy and the adoption of smart grids; social media requires the management and protection of online users' reputation; and online elections cannot succeed without proper management of votes [4].

There are some common operations that can be performed on digital assets to augment their usability. For example, digital assets are transferable across different entities and users via atomic online transactions, such that they are either executed as one unit or not at all. These transactions can take place during a transfer between two bank accounts, a record of fund movement between a sender and a recipient, or a purchase of merchandise using a credit card. Likewise, the management of a special digital asset—digital identity—is important to match these identities in various occurrences to reduce the replication of data. Yet another operation that may be exercised on digital assets is provenance tracking, in which a digital history can be provided for physical products (such as supplies or hardware components) to trace and verify their origins, attributes, and ownership [5].

In recent years, Distributed Ledger Technology (DLT) has emerged as a means to comprehensively capture the advancements of blockchain technologies and variations that extend its core principles [6]. Blockchain technologies today typically refer to decentralized ecosystems managed by consensus mechanisms where the majority of parties (i.e., more than 50%) eventually agrees to the same reality. In such decentralized ecosystems, all the data (i.e., transactions of digital assets) are structured as a chain of blocks and replicated across all network maintainers (miners) [7], just like the Bitcoin blockchain [1].

DLT is an umbrella term that defines any shared ledger (regardless of its internal data structure) maintained in a decentralized network that replicates identical copies of the data across multiple nodes residing in various geographic locations. Nodes in the network simultaneously reconcile their copies of the data through consensus to achieve a shared truth, such that data in the shared ledger is verifiable and tamper-aware. Key to the success of DLT is the consensus process that helps order all valid transactions in a deterministic manner.

In distributed computing, consensus is a mechanism that helps a distributed network establish agreement on the value of some shared data [8]. A distributed ledger network can deliver a consistent and reliable state, even

in the event that one or more nodes may be unreliable due to corruption or hardware failure. Unlike a typical centralized system—where decisions are dictated by the single governing authority—decisions regarding data stored in a distributed ledger are collaboratively made by majority votes. Due to the increasing interest in distributed ledger designs, various consensus mechanisms and information security protocols have surfaced as common configurations employed in these designs. This chapter provides a survey of popular technologies in use or proposed as part of many popular distributed ledgers.

The remainder of this chapter is organized as follows. Section 1 provides an overview of two main types of consensus mechanisms—Byzantine and Non-Byzantine consensus mechanisms—implemented by various distributed ledger systems; Section 2 describes four popular Byzantine consensus mechanisms implemented by public blockchain networks; Section 3 presents three Non-Byzantine consensus mechanisms used by other types of DLT that are permissioned; Section 4 provides an overview of information security protocols implemented in DLT that drive its successful operation; and Section 5 presents concluding remarks.

2. Consensus mechanisms overview

This section provides an overview of two types of mainstream consensus mechanisms—Byzantine and Non-Byzantine consensus mechanisms—implemented by classic distributed systems, which are the foundational technology underlying DLTs today.

2.1 Byzantine consensus mechanisms

Consensus mechanisms in distributed systems can be divided into two categories based on whether they assume maliciousness among the agents. In a classic 1982 paper [9], Lamport, Shostak and Pease introduced the problem of achieving consensus under malicious failure scenarios. They used the example of the Byzantine army and the problem of reaching agreement (consensus) to attack or retreat to explain the problem. Using a basic setup of three generals, they showed that if even one of the generals became malicious the other two generals will not be able to reach consensus. In a general setting, they showed that if at most N generals are traitors, then at least $3N + 1$ generals are needed to ensure that all non-malicious generals ($2N + 1$)

agree on the decision as to whether to attack or retreat. This result has several interesting applications in distributed systems, specially providing a lower bound on the minimum investment needed to ensure fault-tolerance.

2.2 Non-Byzantine consensus mechanisms

In non-malicious (i.e., Non-Byzantine) failure scenarios the problem is to ensure convergence on agreement of a stated value or a sequence of actions even if certain nodes fail. Regardless of whether failures are fail stop or fail stuck, the state is consistently observed by all participants and they agree that the node has failed with the correct stop or stuck semantics. Conversely, in malicious (i.e., Byzantine) cases, a failed node can deceive the other agents into different observations, i.e., some nodes might receive one message from the failed node, whereas other nodes might receive a different message from the failed node.

2.3 Taxonomy of consensus mechanisms

This chapter presents and evaluates several consensus mechanisms implemented by popular public blockchains and other types of permissioned distributed ledger systems. Our evaluation focuses on the following criteria:

- *The degree of decentralization*: the number of miners, maintainers, and/or members allowed in the network
- *Scalability*: transaction throughput, in terms of the time taken for network nodes in a distributed ledger system to reach consensus over a number of transactions grouped in a block or how much time it takes for a block to be produced and accepted by majority nodes
- *Randomness in block generation and miner selection*: dependencies in mining hardware, stakeholding, impact, and importance to the network
- *Consensus type*: whether the consensus mechanism employed by a distributed ledger network is Byzantine or Non-Byzantine consensus in terms of their resiliency to attacks from malicious nodes in the network.

3. Consensus mechanisms used in public blockchains

Public blockchains underlie the vast majority of cryptocurrency-based platforms, such as Bitcoin [1], Ethereum [10], and Litecoin [11]. These types of blockchains are permissionless, decentralized computing architectures open to the public and maintained by arbitrary users who possess Internet

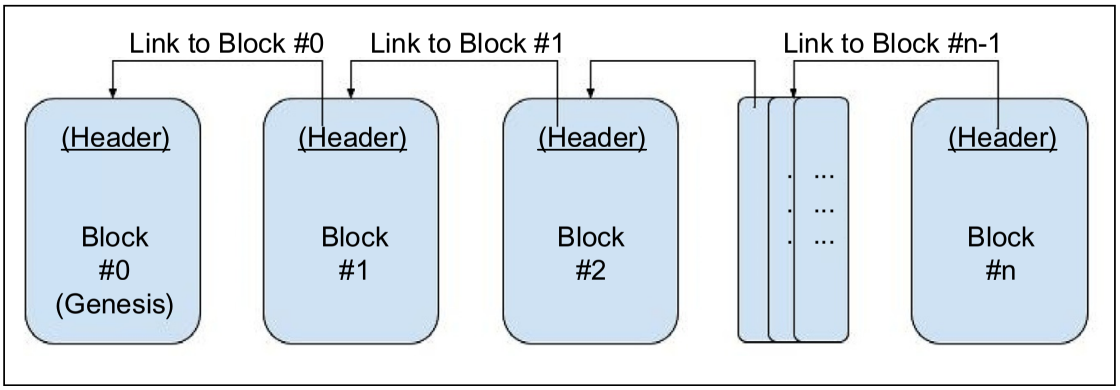


Fig. 1 Blockchain structure: A continuously growing chain of ordered and validated transactions.

access. Anyone with such access can participate in the exchange of digital assets in these platforms. Users are incentivized to contribute to the networks by validating transactions in the hope of being rewarded with digital tokens that may be used for commodity trading or in a shared market. Users are also attracted to public blockchains due to their "trustless" [1] nature, i.e., users can remain anonymous on-chain to protect personal identities and can feel confident that their transactions are carried out with integrity without having a trust relationship with any parties or brokers.

The most common public blockchains are structured as an append-only ledger of transactions that are continually reconciled and verified via a process called blockchain "mining." After a set of transactions is verified by the majority of the network maintainers, the transactions are then grouped into a structured block as being successfully mined. Consequently, the newly mined block is chained to the previous block in the sequence to persist a consistent and ordered transaction history, as shown in Fig. 1.

Public blockchains must guarantee that the shared ledger of transactions always provides the same snapshot to whoever accesses the chain at a given time to avoid incurring large volumes of digital asset exchanges. As a result, public blockchains typically implement the most robust mechanisms to reach consensus in highly decentralized global networks. These mechanisms may be more time consuming than others used in permissioned blockchains, but they more resilient to attacks from (minority) rogue players (in this context, we do not consider the 51% attack where the majority of network nodes collude to reverse blockchain transactions). Transactions in public ledgers are therefore immutable and transparent. Next, we present four consensus mechanisms that have been implemented by popular public blockchains.

3.1 Proof of Work (PoW)

Proof of Work (PoW) was the first prominent blockchain mining mechanism presented in the literature used by the Bitcoin blockchain [1]. With PoW, as new, unverified transactions become available or broadcast to the entire blockchain network, each node that maintains a copy of the ledger (also known as a "miner") verifies a set of those transactions by balancing the incoming and outgoing digital assets with previously validated transactions to prevent so called "double spending."

The miner next groups validated transactions into a tentative block. Each miner then competes in solving a computationally expensive algorithmic "puzzle" to ensure that their block is valid and that it follows in sequence from the last block in the current chain. Only the winner with a correct answer is privileged to append their block of transactions to the shared ledger and gains a mining reward in cryptocurrency (e.g., Bitcoins). This approach is also how native cryptocurrency tokens are minted.

The computationally expensive puzzle is at the heart of the PoW mechanism: it must be hard enough to solve to disincentivize attackers who intend to pollute the blockchain due to the high costs in obtaining a solution. Likewise, validating the proposed solution must be trivial so that it can be easily accepted by other nodes and the solution's correctness is transparent to the network, regardless of the computational power any network node possesses. The puzzle used by Bitcoin is to find a value called a "nonce." This nonce is created by combining the content in the proposed block to produce a new hash output that falls within a target range, such as a target hash prefixed with a number of 0's.

Due to the nature of hashing algorithms, the desirable output of a nonce can be computed only by brute force, i.e., guessing each nonce one by one until a solution is found. It is therefore highly unpredictable which node can successfully mine the next block, thereby protecting the validated transactions from tampering. The puzzle is adjusted regularly to maintain the same level of difficulty. For instance, the puzzle used by Bitcoin results in an average block formation time of 10 min. Fig. 2 illustrates the iterative process of solving the puzzle.

PoW has successfully sustained and secured the operations of two of the most popular public blockchains–Bitcoin and Ethereum–because it helps deter attackers with its requirement of expensive computational power and information transparency. In addition, the cryptocurrency reward incentives for competing to solve the puzzle makes it hard for a small group of rogue actors to manipulate and overpower the majority network nodes.

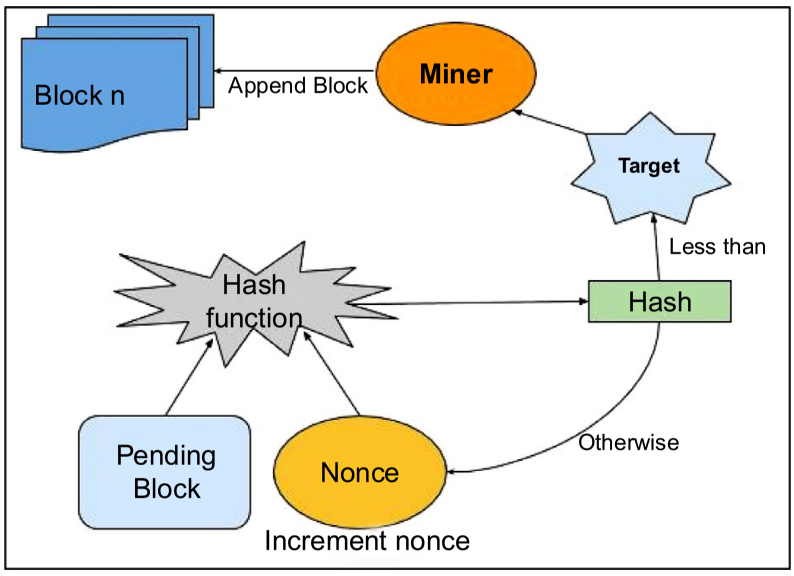


Fig. 2 The iterative process of solving the Proof of Work puzzle.

In practice, however, the computation power required to perform PoW is highly controversial because of its excessive energy consumption and wastage. Another major concern about PoW is its security vulnerability to 51% attack in small-scale public blockchain networks, where not many nodes compete in the mining process. An attacker could exploit those networks by much more easily obtaining a majority of the network's computation power and revert transactions.

3.2 Proof of Stake (PoS)

Another consensus mechanism that has frequently been compared with PoW is Proof of Stake (PoS). To reduce the energy consumption problem introduced by PoW's need for miners to solve a computationally expensive puzzle, PoS determines the next eligible block to append to the chain based on the current "stakes" held by the accounts, i.e., the total native cryptocurrency tokens they have. Stakeholders who are selected to maintain the PoS network are often required to lock in their stakes for a period of time during their service. This nature of PoS provides incentives for nodes to correctly create and validate blocks because by committing to network maintenance with their own shares of tokens, they could risk losing their stake and be deprived of their future privilege as a block producer if they are dishonest. All the locked shares are returnable to the good and fair stakeholders. Moreover, if a block is successfully appended to the blockchain its validator

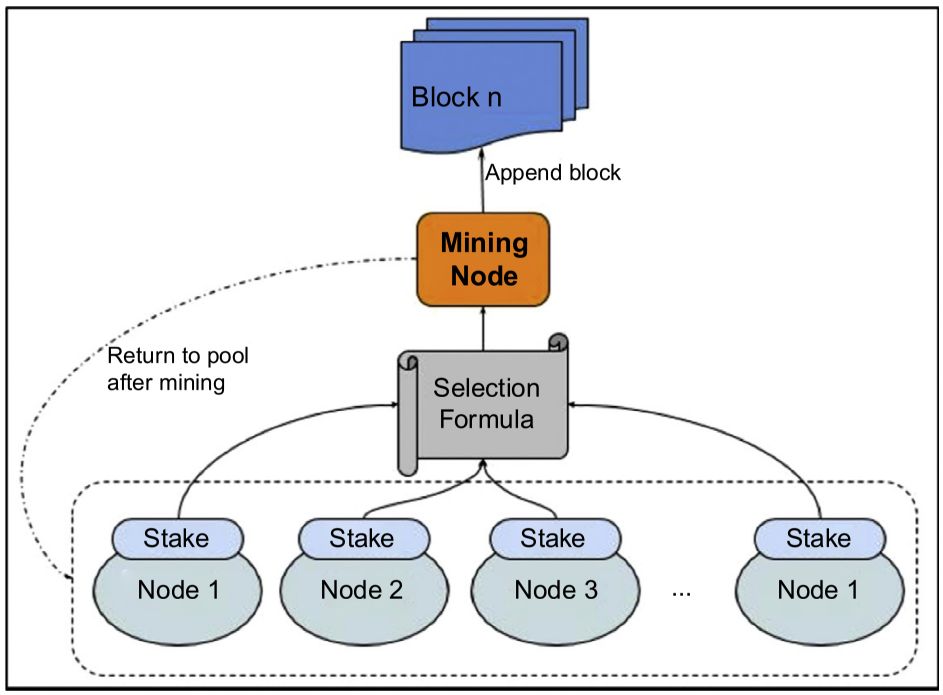


Fig. 3 The iterative process of selecting the next miner to produce a block based on node stakes in the network.

is rewarded with some transaction fees [12]. Fig. 3 summarizes the basic process of PoS-based mining.

Cryptocurrency tokens can typically only be created when the platform is initially launched and/or through the PoW mining process in the early phase (then switching to using PoS) as each new block of transactions is added to the blockchain. In PoS, networks nodes holding more stakes in the network are generally allowed to produce more blocks than others, though the percentage of blocks they are allowed to create are weighted according to the percentage of stakes they own over the entire network. PoS, however, does not simply select the next block based on its validator's cryptocurrency balance to avoid centralizing the network by permanently favoring those with more tokens. Instead, various methods have been proposed to select the next valid block in PoS.

One such method implemented by Nxt [13] and BlackCoin [14] uses randomization in the block selection process to create a formula that calculates the next block based on both the stake of a block's validator and the hash output from its validation. It is possible to predict the next block because all account balances in the shared ledger are available to the entire network. Another technique used in PeerCoin [15] leverages the concept of "coin age" to generate the next block according to both the amount and age

of tokens available in the user accounts. To reduce the chance that one or a small group of users gains advantages due to high stakes in the system, after an account has generated a block, its coin age will be reset to 0 and the counting will restart until a predefined minimum age requirement is met again, e.g., 30 days in PeerCoin.

The main advantage of PoS is energy-efficiency because there is no need for block generators to perform computationally intensive tasks. Likewise, PoS incur lower requirements on computing hardware for users to partake in the block generation process. However, because PoS determines the block sequence according to the wealth of the network maintainers, stakes in cryptocurrency must be already established previously through other means–either minted in a PoW-based system prior to transitioning to PoS or acquired from other users with pre-established stakes. Moreover, because high stakes correspond to more rights to producing blocks, this model may discourage token distribution as users will likely want to keep their tokens instead of spending them.

3.3 Delegated Proof of Stake (DPoS)

The Delegated Proof of Stake (DPoS) consensus mechanism is an increasingly popular alternative to PoW and was first introduced by the founder of BitShares [16] to improve network efficiency and scalability over Bitcoin's PoW mining. DPoS essentially implements a reputation system with voting and an election process among stakeholders to reach network consensus in a digitally democratic manner. Generally, stakeholders in a DPoS-based blockchain system vote for super-representative roles, such as "Witnesses" and "Delegates", as a relatively small (compared to the total number of users with stakes in the network) and fixed number of people to perform critical tasks like validating transactions and generating blocks. Each stakeholder has a number of votes proportional to the tokens they own and may choose to delegate another stakeholder to cast their votes on their behalf [17].

The voting process selects the top N delegates, or also called "Witnesses" in Bitshares, (N being agreed upon by the network to ensure decentralization, e.g., 100 witnesses) based on the total votes received. Witnesses take turns producing and validating blocks and an even smaller number of Witnesses (such as 20) are rewarded with transaction fees for their service. This approach creates competitiveness of the role and deters fraudulent behavior. Voting is also an ongoing process such that any malicious or poorly-performing (due to missed blocks) witness can be voted out of their role at any given time.

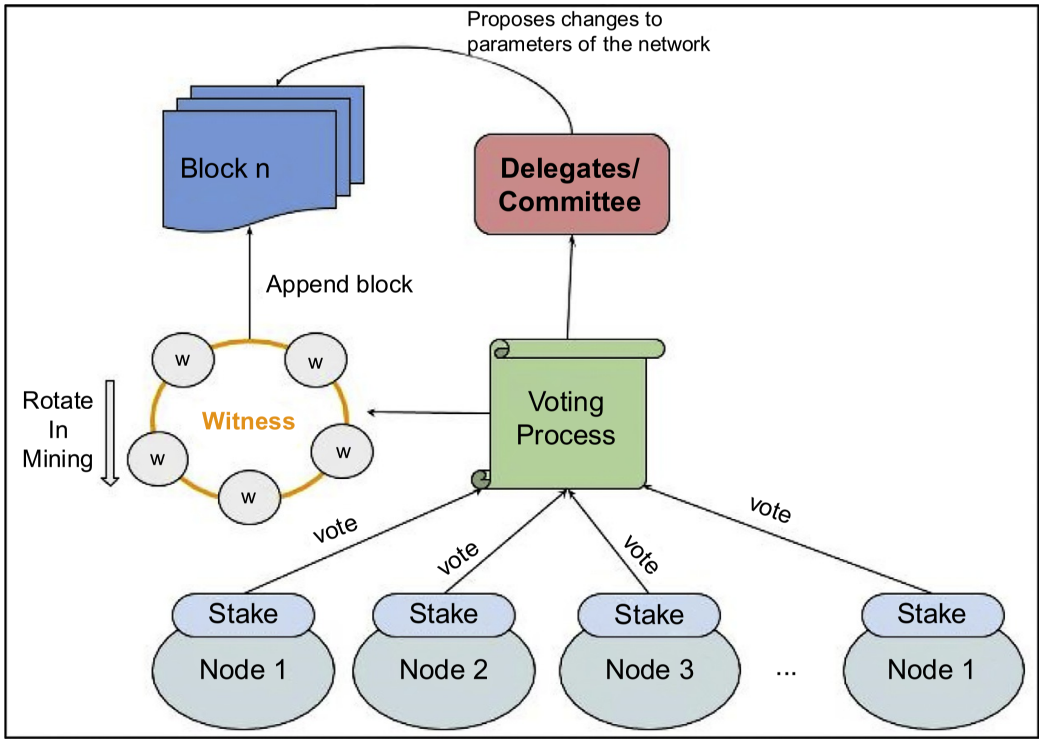


Fig. 4 Nodes use their stakes to vote for "Witnesses" to produce blocks and "Delegates/ Committee" to propose changes to the network parameters.

Another group of delegates, also known as the "Committee" in Bitshares, is also elected to propose necessary changes to network parameters, such as fees paid to Witnesses, block sizes, block intervals, etc. When the majority in the Committee approves a change proposal, the stakeholders then review the proposal and vote to accept or nullify the change and/or vote out a Committee member. Ultimately, the administrative power is distributed to all the stakeholders; super-representatives are not meant to have direct authority to change the network by themselves. The architecture of DPoS is presented in Fig. 4.

A number of popular cryptocurrency networks have successfully implemented DPoS, such as BitShares [17], EOS [18], Steem [19], and Cardano [20]. Similar to PoS, DPoS offers more efficient transaction processing and a much lower requirement on computing hardware than PoW. In its existing deployments, DPoS has created more decentralized networks than its PoS predecessor, owing to its continuous election process that involves users even with the minimum token ownership. The downside to DPoS is its long-term sustainability as its governance relies heavily on users to actively elect a small set of delegates, which may be vulnerable to centralization overtime due to smaller stakeholders forfeit voting rights and/or tokens become poorly distributed.

3.4 Proof of Importance (PoI)

Proof of Importance (PoI) is a consensus mechanism introduced in the NEM blockchain platform [21]. It resembles the stake ownership consideration in PoS and DPoS since users are only eligible to forge (or "harvest" as in NEM) a block if they meet a minimum requirement of stakes in the network's native cryptocurrency. For instance, NEM currently only selects candidates from a pool of users with at least 10,000 XEM vested. PoI differs from all the aforementioned mechanisms, however, since it takes into account other factors than the amount of computation one puts into or the amount of stake one holds alone. In particular, PoI determines a user's eligibility of harvesting blocks according to their overall contribution to the network, rather than a single aspect, using a more holistic metric named the "importance score". Users having a higher importance score are more likely selected to harvest a block and are rewarded with transaction fees for their work.

Upon meeting the minimum stockholding requirement, PoI calculates an importance score for each user using a heuristic function based on the following three factors:

- *The user's token ownership*, where more tokens will correlate to a higher importance score as long as the tokens have been available in the user's account for a fixed period.
- *The net outbound token transactions*, which rewards the distribution of cryptocurrencies instead of accumulating or concentrating wealth. Calculating net transfers prevent accounts from gaming the system, such as transacting between different personal accounts.
- *The connectivity to other network nodes*, which promotes account interactions across the network by assigning more importance to more active users with diverse transaction history and larger transaction amount [22].

Fig. 5 presents an overview of the PoI architecture.

PoI was designed to minimize the centralization of wealth as a potential disadvantage of PoS. In particular, it uses the importance score to stimulate the spread of wealth by giving high scores to users with larger transactions and more/or connected transaction networks. PoI is intentionally resilient to the "nothing at stake" problem, where PoS valuators maintain and hold stakes in *every* fork encountered in the network because the costs to do so are trivial given that minimal external resources are needed, unlike in PoW. In this hypothetical scenario, a malicious attacker could exploit a double spend by using the current main chain and his/her fork. Since all forks are built

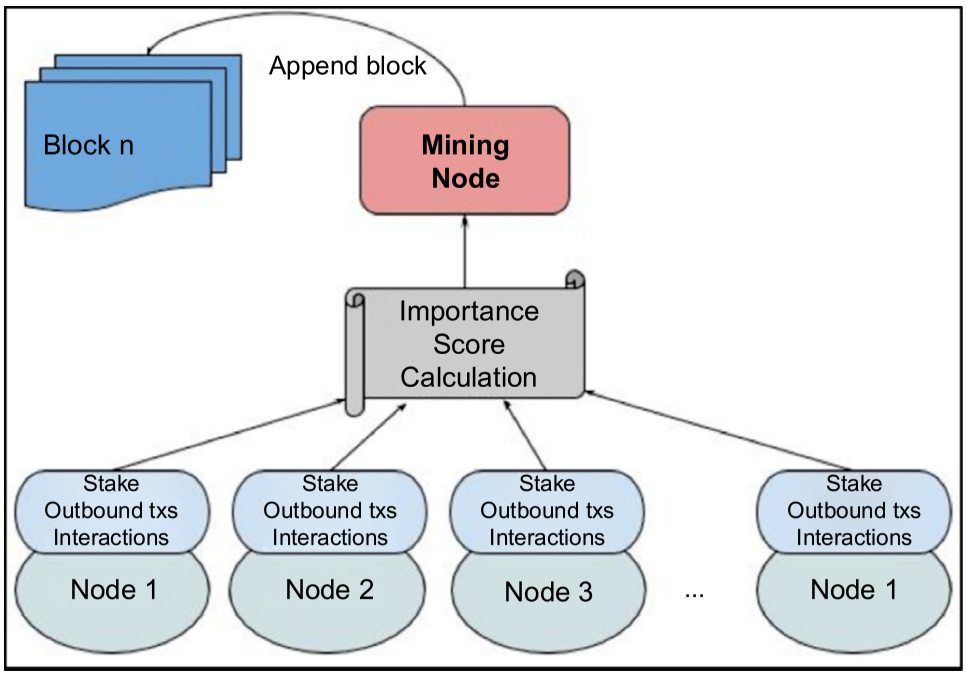


Fig. 5 An overview of the architecture of PoI consensus that calculates an importance score used to select the next mining node.

simultaneously an attacker could simply choose his/her fork containing the invalid transaction as the main chain and succeed in the attack when the stakeholding attacker is selected as the next block producer.

Despite the importance score taking into account several aspects of a user's involvement in and contribution to the network, it may be shifted towards more centralization in the longer term because the calculation still favors wealthy users with more stakes in the network and thus more flexibility to transact and interact with other users.

4. Consensus mechanisms used in other forms of distributed ledger technology

Public blockchains are designed to maximize the level of transparency and decentralization to provide a trustless environment for users interested in exploring the network and/or actively exchanging digital assets (such as cryptocurrencies) freely and anonymously. The openness and lack of restrictions on data access, however, may not be ideal for entities or functions that require sensitive data warehousing or exchange, such as enterprises or government agencies. Moreover, consensus mechanisms implemented in public blockchains must exercise strict orders to protect the networks, rendering transactions relatively slow compared to industry requirements.

To address these limitations, therefore, permissioned blockchains and other variants of distributed ledger technologies have surfaced as more closed ecosystems that restrict access and loosen some constraints of the consensus requirements of public blockchains. Permissioned blockchains are similar in structure to public blockchains, while permissioned distributed ledgers may store transactions in a single linear chain of blocks, multiple chains of blocks, or a directed acyclic graph (similar to a tree structure) that is non-linear [23].

Permissioned distributed ledgers allow their members to limit user access to the network by disclosing their identities prior to joining the consortium. They establish a much smaller and more controlled environment that has a modicum of trust among the member nodes. Consensus is therefore much easier and faster to achieve in permissioned networks compared to ones completely open to the public. Despite the lower degree of decentralization, consensus is still a crucial process to ensure that every member has equal rights to updating the shared ledger in the network. Below we describe three popular consensus mechanisms used in permissioned ledgers. Table 2 presents a summary that compares these mechanisms.

4.1 Proof of Elapsed Time (PoET)

In Proof of Elapsed Time (PoET), nodes are selected to produce blocks after waiting for a random period of time. This technique was developed by Intel and has been adopted by the Hyperledger Sawtooth project [24] as a much leaner alternative to PoW in a permissioned network [25]. Its core mechanism is based on Intel's Software Guard Extensions (SGX) technology [26] that has the ability to digitally attest that some code has been correctly set up in a so called "Trusted Execution Environment" [27]. In PoET, this code is a function that generates a random time period that must be waited out by each node.

When a participant joins the network, they download the time generator code and receive an attestation (in the form of a digital certificate) of the code setup from SGX that they announce to the network. Existing members can either approve or deny the join request. If the request is approved, the new node becomes an eligible candidate to produce blocks and participates in the random selection process. Whoever first completes the waiting period broadcasts a signed message to the network as the randomly chosen next block forger. The SGX is a critical component because (1) it warrants the integrity of the randomly generated wait period, preventing malicious nodes

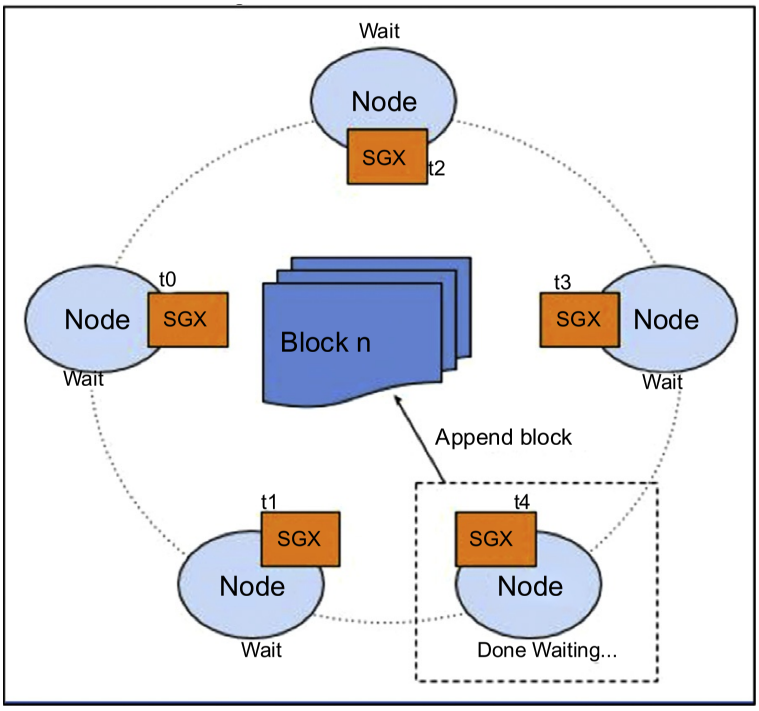


Fig. 6 Architecture of the PoET consensus mechanism that leverages the Software Guard Extension (SGX) technology to select the next miner based on a random wait period.

from altering the timer code to their advantage and (2) the attestations are an efficient method to verify the validity of wait time completion [25]. The PoET consensus mechanism is illustrated in Fig. 6.

PoET is an efficient and scalable mechanism, especially for permissioned network. It creates a randomized model for selecting block producers without resource-intensive computing as in PoW systems or complex calculations for determining miners used in consensus mechanisms involving stakes in the system as in PoS and PoI. However, PoET heavily relies on Intel's specialized, third-party hardware to operate, which creates entry barriers for participants without access to the SGX technology. It is also possible for nodes with more hardware available to gain a better chance of getting selected, but such nodes may likely be denied access to join the permissioned network.

4.2 Proof of Authority (PoA)

Proof of Authority (PoA) is designed to optimize the PoS mechanism and be used, ideally, in permissioned networks. Instead of choosing block miners on the basis of their stakes in cryptocurrency tokens, PoA selects a small group of authorities as transaction validators by their identity or reputation staked

in the network [28]. To contend for validators, users go through a formal notarization process in which they provide documentation to prove their real identities and link them with their on-chain identities to establish their digital reputation. Existing validators can vote to add additional users into the authority group. A PoA-based system also rewards authorities for certifying and ordering transactions to incentivize honest behavior in providing service and moderating the network. PoA Network [29] and Ethereum's test net Kovan [30] are examples of public networks that use PoA consensus. An overview of the PoA mechanism is shown in Fig. 7 below.

PoA does not require intensive computation to complete hard tasks and only relies on a small number of validators to reach consensus. These features help improve transaction throughput and energy efficiency compared with PoW- and PoS-based systems. However, PoA also forgoes decentralization by concentrating mining power among a group of trusted authorities. As a result, this model can introduce censorship into the public network where one or more authorities may blacklist or deny all transactions from a particular user. On the other hand, a permissioned network established between different enterprise or large institutions can benefit from PoA because it offers a faster transaction processing speed and the identity-at-stake model aligns well with business operations that value trustworthiness and reputation.

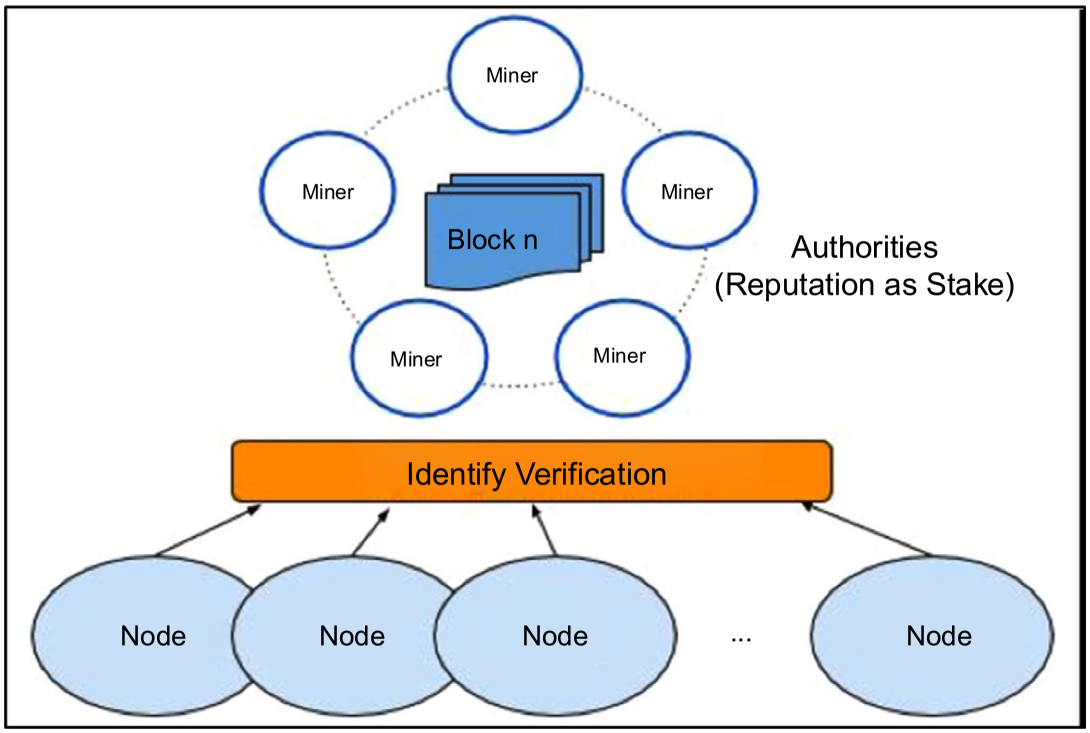


Fig. 7 Architecture of the PoA consensus mechanism in which nodes put their reputation at stake by verifying their identities before becoming miners in the network.

4.3 Ordering-based consensus

Ordering-based consensus is commonly applied in permissioned networks in which nodes are selected to participate in the network that supports a membership or identity service. A well-known permissioned ledger system, Hyperledger Fabric [31], implements a three-step process to reach consensus in this manner. First, the client application proposes a transaction to a set of nodes called "peer nodes" (who hosts of the shared ledger) for endorsement based on some predefined policy (e.g., requiring M out of N signatures from peer nodes). Peer nodes next return a response of the transaction proposal to the client, who then submits the endorsed transaction to a node called the "orderer." Orderer nodes may receive endorsed transactions in different orders, but they collectively determine the final, strict sequence of the transactions and package them into immutable blocks. Finally, the orderer nodes distribute transaction blocks to connected peer nodes for validation of the new transaction blocks. In this model, if a transaction is deemed invalid, it remains in the block, but is marked as invalid by the peer node because blocks created by orderer nodes are in their final states [32]. Fig. 8 presents the high-level architecture of order-based consensus.

The ordering service is key to reaching consensus regarding the states and sequence of transactions in the ledger. One of the consensus mechanisms from distributed computing that is implemented by the ordering service

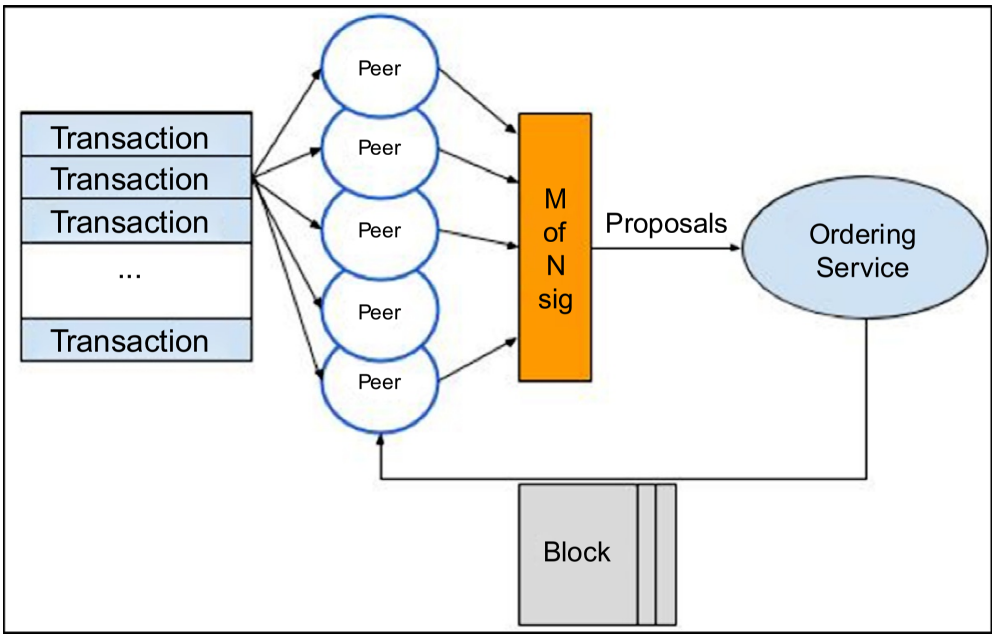


Fig. 8 Overview of an order-based consensus architecture that relies on an ordering service to determine sequences of transactions.

in Hyperledger Fabric is a leader-follower model called Raft [33]. In Raft, a single leader node is elected in each "term" of an arbitrary length to make decisions and propagate those decisions to the followers who then replicate them. Each Raft node maintains a log and is always in one of three states: follower, candidate, or leader. A node starts as a follower who accepts and replicates log entries from the leader. If a leader is not present or has not been responsive for a time period, the follower waits for a randomized timeout period and then moves into the candidate state. The candidate then requests votes from other nodes and becomes the leader if it receives majority votes from the network. All new transaction entries go through the leader, who appends the changes to its log and replicates the effective transaction sequence to the followers. The leader waits until a majority of followers have updated their local logs and then commits the updates and broadcasts the confirmation to follower nodes. The network is now in consensus of the transaction sequence and ledger states [34].

The ordering-based consensus process designates separate roles for verifying and ordering transactions. The leader-follower model orders transactions at a fast pace because the order is determined effectively by a single leader node at a time instead of being computed by every node. It does require configurations to set up the architecture but not specialized hardware or extensive computing resources, making it a cost-effective model for permissioned ledger systems. The leader election process can quickly detect a faulty leader in the network and replaces it so that the ledger can be updated continuously as new transactions are received. Raft-based ordering service alone, however, is not resilient to attacks from malicious nodes that may exist within the network, which requires other components in place to validate transactions added by leader nodes, as are present in the Hyperledger Fabric system.

5. Information security technologies

The success of distributed ledger technology is driven by a combination of

- core computer science concepts and principles from distributed systems, which are key to the development of consensus mechanisms, and

- information security, which ensures the security and integrity during transactions of digital assets.

Below we present five important concepts from information security commonly integrated into distributed ledgers.

5.1 Public key cryptography (encryption and signing)

Public key cryptography [35] is a critical component of distributed ledger technology. It is composed of a pair of mathematically related public and private keys generated from one-way cryptographic function. While the public key can be freely distributed, the private key must be protected by the owner of the key pair. It is computationally infeasible to acquire the private key given its paired public key in the cryptographic keys commonly used in information security today.

A private key is used commonly in public key encryption to safeguard the secrecy of messages using the intended recipient's public key to encrypt data packets, ensuring that only owners of the corresponding private key can decrypt them. Another standard use case is for the sender of data to create digital signatures with their private key, with the signatures being easily verifiable using their public key, to prevent data from being tampered with in transit. Through public key cryptography, public blockchains like Bitcoin and Ethereum provide so-called "pseudo-anonymity" [1], meaning that as long as users' real-life, personal identity is not linked with their public keys on-chain, their activities can remain anonymous.

5.2 Hashing

Hashing is a critical process employed by distributed ledgers to map data of an arbitrary length (e.g., different types/sizes of digital assets exchanged) to data of a fixed size (e.g., transaction hashes stored in the shared ledger) using cryptographic functions [36]. These functions are one-way, meaning that it is trivial to generate a hash value from a given input but impractical (given computation capabilities today) to reverse engineer and calculate the input value. Given the same input, hash functions will always produce the same output. Moreover, even the slightest change in the input will completely alter the output. Hashing is commonly used in blockchains to protect transacted data against tampering and link new validated transactions with the existing ledger to create a network-wide non-refutable history.

5.3 Multi-signature

Multi-Signature (multisig) is a joint digital signature created by more than one party to improve the protection and integrity of the original content [37]. The goal of this approach is to more securely authenticate transactions than the traditional single public-private key pair. Multisig is often employed by cryptocurrency wallets or public blockchain networks.

In early cryptocurrency-based networks, a user whose private key has been lost or stolen could permanently lose access to the ownership of their digital assets. Moreover, cryptocurrency accounts created for business operations are extremely prone to insider attacks because anyone with access to the shared secret key can withdraw or transfer the balance without being traceable. Multisig is designed to overcome these issues by requiring more than a single key pair to authenticate transactions. It requires M keys out of a set of N ($N >= M$) key pairs to perform a transaction, with 2 out of 3 keys ($M = 2$, $N = 3$) being the most common scheme today.

5.4 Ring signature

Despite the use of public key cryptography, logs in the shared ledger can be traced to identify certain patterns in users especially when users execute repeated transactions or interact with the same set of other users. The traceability of transactions is not ideal for privacy-focused blockchains that seek to obfuscate the identity of users originating the exchange of digital assets [4]. Ring signature is one method applied in Bytecoin [38] and Monero [39] to protect on-chain privacy for the sender of a transaction.

In ring signature, the actual signer of a message forms a group with an arbitrary number of other users or decoys, each of which has a public-private key pair. The signer then produces a ring signature and a one-time random ring key pair using a series of mathematic calculations based on a combination of the message, his own secret key, and all other users' public keys. The signature is verifiable using all the public keys from the ring. To an outside observer, the actual signer is computationally indistinguishable from other parties in the ring, and therefore, the identity of the signer is no longer traceable [40].

5.5 Zero-Knowledge Proof (ZKP)

Concerns about data privacy in shared environments are arising as distributed ledger technology is increasingly touted as a decentralized data transaction infrastructure that removes centralized control, in popular

domains, such as finance [41,42], supply chain [43,44], and healthcare [45,46]. Vital information that could be used to identify an individual, such as date of birth, social security numbers (in the U.S.), employment information, and bank statements, is paramount to the safety and financial well-being of the identity owner. To safeguard sensitive information, initial applications of zero-knowledge proof (ZKP) techniques have surfaced in DLT projects like the zk-SNAKRS [47,48] protocol in ZCash.

ZKP is a complex scheme designed to incorporate encryption techniques to enable a prover to certify the truthfulness of a statement to a verifier without disclosing any more specifics other than the statement itself. A true ZKP must possess the following three key properties:

- *Completeness*—if the statement is true, an honest prover will convince the verifier,
- *Soundness*—if the statement is false, verifier will find out the prover is dishonest with very high probability, and
- *Zero knowledge*—if the statement is true, no extra information is revealed to the verifier other than the statement being true [49].

6. Concluding remarks

This chapter provided a survey of various consensus mechanisms and key information security concepts employed in public and permissioned distributed ledgers for exchanging and distributing digital assets. Each consensus mechanism can optimize at most two of the three attributes in a DLT network described in Section 2.3: *degree of decentralization* (number of network miners/maintainers/members), *scalability* (transaction throughput; number of transactions per second), and *randomness in block generation and miner selection* (dependencies in mining hardware, stakeholding, impact and importance to the network). Tables 1 and 2 provide an overview and summary of the consensus mechanisms described and compared in this chapter.

Generally, public blockchains require Byzantine consensus to maintain a robust and resilient decentralized network since any node, including malicious node, can theoretically become a miner. Permissioned networks, in contrast, are often less concerned with malicious nodes because strict rules are in place to onboard members to ensure that nodes in the network can be trusted with their reputation stake in the network.

Table 1 A comparison of consensus mechanisms used in public blockchains.

Consensus mechanism	Degree of decentralization	Scalability	Randomness in miner selection	Consensus type
Proof of Work (PoW)	High—allows any node to join the network and become a miner in the network; although as hardware requirement increases, mining power may become less decentralized	Low—transaction throughput is low due to difficulty in solving the cryptographic puzzle	High—cryptographic puzzle can only be solved by brute force; difficult to know which miner will solve it first despite nodes with more computational power having higher chances of winning	Byzantine consensus
Proof of Stake (PoS)	Medium—allows any node to join the network but only node with higher stakes in the network can become a miner; the stake holding requirement may eventually regionalize or centralize power	High—no requirement on solving a difficult puzzle to reach consensus	Medium—nodes with higher stakes in the network are more likely chosen as miners	Byzantine consensus
Delegated Proof of Stake (DPoS)	Medium—allows any node to join the network, but only a small subset of nodes is selected as miners	High—no requirement on solving a difficult puzzle to reach consensus	Low—although the election of miners is fairly random by majority votes, chosen miners take turn to generate blocks	Byzantine consensus
Proof of Importance (PoI)	High—allows any node to join the network and become a miner in the network, but the barrier of entry into the miner pool is high as initial stakes are required	High—no requirement on solving a difficult puzzle to reach consensus	Low—mining eligibility is highly dependent upon the calculation of an importance score, which can be highly predictable	Byzantine consensus

Table 2 A comparison of consensus mechanisms used in other permissioned distributed ledger networks.

Consensus mechanism	Degree of decentralization	Scalability	Randomness in miner selection	Consensus type
Proof of Elapsed Time (PoET)	Low—requires miners to possess specialized hardware	High in terms of transaction throughput—can produce transactions at a very fast speed without the need to solve a hard puzzle	High—specialized hardware used generates a random timer used to determine the next miner; the process is also verifiable	Non-Byzantine consensus
Proof of Authority (PoA)	Low—theoretically, anyone who is willing to go through an identity verification process could be appointed as an authority to generate blocks; in reality, a relatively small number of people will be appointed	High in terms of transaction throughput—a small set of authorized nodes are responsible for processing transactions and generating blocks	Low—the identity verification process most likely selects miners with established reputation	Non-Byzantine consensus
Ordering-based consensus	Low—resilient to faulty nodes but not to attacks from malicious nodes that may exist in the network; a tighter control would therefore be needed to prevent against those attacks	High in terms of transaction throughput—a single leader orders transactions at a time and replicates the changes to all follower nodes	High—the leader selection process is based on a randomized timeout period	Non-Byzantine consensus

Key terminology and definitions

Consensus A mechanism to achieve overall reliability in a distributed ledger network in the presence of a number of faulty maintainer and mining nodes. It requires the majority (51%) of nodes to agree on content stored in the shared ledger.

Delegated Proof of Stake A scalable consensus mechanism that uses a democratic voting and election process to select delegates who create and validate blocks of transactions; the number of votes a user can cast is proportional to the tokens they own; delegates are incentivized to be honest with rewards and also the risk of being voted out.

Hashing A cryptographic process that maps an input data of variable length to an output with a fixed size using a one-way function such that it is computationally infeasible to calculate the input based on the output; even the slightest changes to the input data will alter the data output; it is used to create transactions that are stored in shared ledgers to protect the content of transactions from being exposed or tampered with.

Multi-Signature (Multisig) A digital signature jointly created by more than one party to provide more security and integrity protection during an exchange of digital assets; it usually follows a scheme that requires M out of N signatures, where $M <= N$ and $N >= 2$ such that a single party alone cannot sign a message or authorize a transaction; it is used by many cryptocurrency wallets to prevent frozen funds due to a single private key being lost or stolen.

Proof of Elapsed Time An efficient consensus protocol designed for permissioned distributed ledger networks to randomize the selection of block miners. It is based on Intel's Software Guard Executions technology that attests for the integrity of some trusted code used to generate a random waiting period for each node. The first node to finish waiting is given the privilege to mint the next block.

Proof of Importance A consensus mechanism introduced in the NEM blockchain that uses an importance score to select block generators, based on stake ownership, the spread of cryptocurrency, and interactions with other nodes to incentivize the distribution and transactions of native tokens.

Proof of Stake An energy-efficient consensus mechanism in which block producers are selected randomly based on their stakes of cryptocurrencies in the blockchain; users can mine a percentage of blocks in proportion to their token balance; malicious behavior is disincentivized because of the risk of losing stakes and privilege to produce blocks.

Proof of Work A robust consensus mechanism often used in public blockchains in which network nodes compete to solve a computationally expensive puzzle, whose solution is trivial to verify, to ensure the validity and integrity of ordered transactions.

Public Key Cryptography Also known as asymmetric cryptography, which is an encryption scheme that uses a mathematically related key pair, a public key and a private key, to secure information. The public key is used to encrypt data, while and the private key is used to decrypt cipher text to obtain the original data. It is computationally infeasible to calculate the private key based on the public key. As a result, public keys can be freely distributed for encrypting content and verifying digital signatures; however, private keys are kept secret with their owners to use for decrypting content and creating digital signatures.

Raft A consensus mechanism that follows a leader-follower model; A single leader is randomly elected to decide upon shared states of the network and broadcasts the changes to follower nodes; the election process that is based on randomized timeout settings

occurs when a leader is not present or has not been responsive for a pre-defined time period.

Ring Signature A digital signature created by a signer who forms a group with other arbitrary users or decoys, each with a public-private key pair; the signature obfuscates the identity of the actual signer by generating a one-time signing key from all members of the group; its application in distributed ledger transactions protects the identity of the sender.

Zero-Knowledge Proof A cryptographic scheme that allows a certifying party to prove to a verifier that a statement is true without disclosing any other information about what the statement is; it allows a secret (such as sensitive information) to be used for verification purposes without the exact detail or specifics to be known to others.

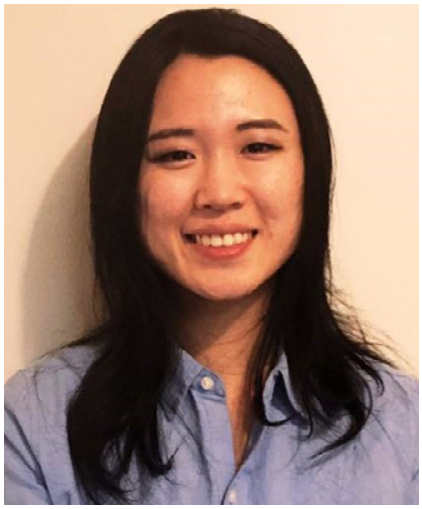
References

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, Bitcoin.org.
[2] N. Cahn, Postmortem life on-line, Prob. Prop. 25 (2011) 36.
[3] E. Keathley, Digital Asset Management: Content Architectures, Project Management, and Creating Order Out of Media Chaos, Apress, 2014Retrieved 04-30-2019.
[4] J.W. TEP, S. Solicitors, Digital assets; a legal minefield, in: Zürich: STEP Verein & Basel Conference, 2014. URL: https://stoanalytics.com/article/digital-assets-a-legal-minefield/. Retrieved 04-30-2019.
[5] T. Regli, Digital and marketing asset management: the real story about DAM technology and practices, Rosenfeld Media, 2016.
[6] S. Davidson, P. De Filippi, J. Potts, Disrupting governance: the new institutional economics of distributed ledger technology, Available at SSRN 2811995, SSRN https://www.ssrn.com/index.cfm/en/, 2016.
[7] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, Blockchain technology: beyond bitcoin, Appl. Innov. 2 (6–10) (2016) 71.
[8] M.J. Fischer, N.A. Lynch, M.S. Paterson, Impossibility of distributed consensus with one faulty process, Massachusetts Inst of Tech Cambridge lab for Computer Science, 1982.
[9] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (3) (1982) 382–401.
[10] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, vol. 151, Ethereum Project Yellow Paper, 2014, pp. 1–32.
[11] "Litecoin v0.16.3," Sept. 29, 2018, URL: https://litecoin.org/. Retrieved 04-30-2019.
[12] V. Buterin, What Proof of Stake is And Why it Matters, vol. 26, Bitcoin Magazine, 2013. August.
[13] "Whitepaper:Nxt(Blocks)", URL: https://nxtwiki.org/wiki/Whitepaper:Nxt, 2018. Retrieved 04-30-2019.
[14] P. Vasin, Blackcoin's Proof-of-Stake Protocol v2, URL: https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf,vol. 71, 2014. Retrieved 04-30-2019.
[15] S. King, S. Nadal, Ppcoin: Peer-to-Peer Crypto-Currency With Proof-of-Stake, vol. 19, Self-Published Paper, 2012.
[16] D. Larimer, Delegated Proof-of-Stake (Dpos), Bitshare Whitepaper, 2014.
[17] "Delegated Proof-of-Stake Consensus" URL: https://bitshares.org/technology/delegated-proof-of-stake-consensus/. Retrieved 04-30-2019, 2019, bitshares.org.
[18] Eos.io, EOS.IO Technical White Paper v2, March 16, 2018, https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md.

[19] Steem.com, Steem An incentivized, blockchain-based, public content platform. August 2017, URL: https://steem.com/SteemWhitePaper.pdf.
[20] "Cardano", URL: https://www.cardano.org/en/home/. Retrieved 04-30-2019.
[21] NEM, February 23, 2018, "NEM—Distributed Ledger Technology (Blockchain) Technology," URL: https://nem.io/technology/.
[22] H. Lombardo, NEM Q&A—Original, Tested Blockchain Platform, Proof-of-Importance, "Change the World, Forever" Tech, 2015. URL: http://allcoinsnews.com/2015/04/07/nem-qa/, Retrieved 04-30-2019.
[23] N. Kannengießer, S. Lins, T. Dehling, A. Sunyaev, What does not fit can be made to fit! trade-offs in distributed ledger technology designs, in: Trade-Offs in Distributed Ledger Technology Designs, 2019 (January 10, 2019).
[24] Hyperledger Sawtooth, 2018, URL: https://www.hyperledger.org/projects/sawtooth. Retrieved 04-30-2019.
[25] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, C. Montgomery, Sawtooth: An Introduction, The Linux Foundation, 2018.
[26] F. McKeen, et al., Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave, in: Proceedings of the Hardware and Architectural Support for Security and Privacy 2016, ACM, 2016, p. 10.
[27] M. Sabt, M. Achemlal, A. Bouabdallah, Trusted execution environment: what it is, and what it is not, in: 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1, IEEE, 2015, pp. 57–64.
[28] W. Gavin, PoA Private Chains, URL: https://github.com/ethereum/guide/blob/master/poa.md, 2015. Retrieved 04-30-2019.
[29] "POA", URL: https://poa.network/. Retrieved 04-30-2019.
[30] "Kovan PoA Testnet Proposal", URL: https://github.com/kovan-testnet/proposal, 2017. Retrieved 04-30-2019.
[31] E. Androulaki, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, ACM, 2018, p. 30.
[32] C. Cachin, Architecture of the hyperledger blockchain fabric, in: Workshop on distributed cryptocurrencies and consensus ledgers, vol. 310, 2016.
[33] The Raft Consensus Algorithm, URL: https://raft.github.io/. Retrieved 04-30-2019.
[34] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14), 2014, pp. 305–319.
[35] W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson, Upper Saddle River, 2017.
[36] A.G. Konheim, Hashing in Computer Science: Fifty Years of Slicing and Dicing, John Wiley & Sons, 2010.
[37] M. Bellare, G. Neven, Identity-based multi-signatures from RSA, in: Cryptographers' Track at the RSA Conference, Springer, 2007, pp. 145–162.
[38] Bytecoin, Cryptography behind Bytecoin, 2018URL: https://bytecoin.org/blog/cryptography-behind-bytecoin, 2018. Retrieved 04-30-2019.
[39] Moneropedia, In: Ring Signature, URL: https://www.getmonero.org/resources/moneropedia/ringsignatures.html. Retrieved 04-30-2019.
[40] R.L. Rivest, A. Shamir, Y. Tauman, How to leak a secret, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2001, pp. 552–565.
[41] B. Scott, How can cryptocurrency and blockchain technology Play a role in building social and solidarity finance? UNRISD Working Paper, 2016.
[42] Y. Guo, C. Liang, Blockchain application and outlook in the banking industry, Financ. Innov. 2 (1) (2016) 24.
[43] K. Korpela, J. Hallikas, T. Dahlberg, Digital supply chain transformation toward blockchain integration, in: Proceedings of the 50th Hawaii International Conference on System Sciences, 2017.

[44] S.A. Saveen, R.P. Monfared, Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger, eSAT, 2016. https://dspace.lboro.ac.uk/dspace-jspui/handle/2134/22625.
[45] P. Zhang, D.C. Schmidt, J. White, G. Lenz, Blockchain technology use cases in healthcare, in: Advances in Computers, vol. 111, Elsevier, 2018, pp. 1–41.
[46] P. Zhang, J. White, D.C. Schmidt, G. Lenz, S.T. Rosenbloom, Fhirchain: applying blockchain to securely and scalably share clinical data, Comput. Struct. Biotechnol. J. 16 (2018) 267–278.
[47] C. Rackoff, D.R. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, in: Annual International Cryptology Conference, Springer, 1991, pp. 433–444.
[48] What are zk-SNARKs?, URL: https://z.cash/technology/zksnarks/. Retrieved 04-30-2019.
[49] O. Goldreich, Y. Oren, Definitions and properties of zero-knowledge proof systems, Journal of Cryptol. 7 (1) (1994) 1–32.

About the authors



Dr. Peng Zhang recently received her M.S. and Ph.D. in Computer Science from Vanderbilt University, Nashville, TN. She previously received her B.S. degree in Computer Engineering from Lipscomb University [2010–2013] also in Nashville. Her research interests include model-driven design for engineering and healthcare IT systems, intelligent model constructions using machine and deep learning, decentralized algorithms and protocols for facilitating and securing clinical communications, and application and enhancement of Blockchain technologies for moving towards patient-centered care. She has interned with industry companies such as HiTactics, Varian Medical Systems, and Center for Medical Interoperability to lead various machine learning and blockchain-related research projects. Dr. Zhang's work on FHIRChain, a blockchain-based architecture for enabling secure and scalable healthcare data sharing has been recently covered by HealthDataManagement, BeckersHospitalReview, HCANews, and several other media outlets.

Dr. Douglas C. Schmidt is the Cornelius Vanderbilt Professor of Computer Science, Associate Provost for Research Development and Technologies, Co-Chair of the Data Sciences Institute, and a Senior Researcher at the Institute for Software Integrated Systems, all at Vanderbilt University. His research covers a range of software-related topics, including patterns, optimization techniques, and empirical analyses of middleware frameworks for distributed real-time embedded systems and mobile cloud computing applications.

Dr. Schmidt has published 12 books and more than 600 technical papers covering a range of software-related topics, including patterns, optimization techniques, and empirical analyses of frameworks and model-driven engineering tools that facilitate the development of mission-critical middleware and mobile cloud computing applications running over wireless/wired networks and embedded system interconnects. For the past three decades, Dr. Schmidt has led the development of ACE and TAO, which are open-source middleware frameworks that constitute some of the most successful examples of software R&D ever transitioned from research to industry.

Dr. Schmidt received B.A. and M.A. degrees in Sociology from the College of William and Mary in Williamsburg, Virginia, and an M.S. and a Ph.D. in Computer Science from the University of California, Irvine in 1984, 1986, 1990, and 1994, respectively.

Dr. Jules White is an Assistant Professor in the Department of Electrical Engineering and Computer Science at Vanderbilt University. He was previously a faculty member in Electrical and Computer Engineering and won the Outstanding New Assistant Professor Award both at Virginia Tech. His research has produced over 85 papers and won 5 Best Paper and Best Student Paper Awards. Dr. White's research focuses on securing, optimizing, and leveraging data from mobile cyber-physical systems. His mobile cyber-physical systems research spans focus on: (1) mobile security and data collection, (2) high-precision mobile

augmented reality, (3) mobile device and supporting cloud infrastructure power and configuration optimization, and (4) applications of mobile cyber-physical systems in multi-disciplinary domains, including energy-optimized cloud computing, smart grid systems, healthcare/manufacturing security, next-generation construction technologies, and citizen science. His research has been licensed and transitioned to industry, where it won an Innovation Award at CES 2013, attended by over 150,000 people, was a finalist for the Technical Achievement at Award at SXSW Interactive, and was a top 3 for mobile in the Accelerator Awards at SXSW 2013. His research is conducted through the Mobile Application computinG, optimizatoN, and secUrity Methods (MAGNUM) Group at Vanderbilt, which he directs.

Dr. Abhishek Dubey is an Assistant Professor of Electrical Engineering and Computer Science at Vanderbilt University, Senior Research Scientist at the Institute for Software-Integrated Systems and co-lead for the Vanderbilt Initiative for Smart Cities Operations and Research (VISOR). His research interests include model-driven and data-driven techniques for dynamic and resilient human cyber physical systems. He directs the Smart computing laboratory (scope.isis.vanderbilt.edu) at the university. The lab conducts research at the intersection of Distributed Systems, Big Data, and Cyber Physical System, especially in the domain of transportation and electrical networks. Abhishek completed his Ph.D. in Electrical Engineering from Vanderbilt University in 2009. He received his M.S. in Electrical Engineering from Vanderbilt University in August 2005 and completed his undergraduate studies in Electrical Engineering from the Indian Institute of Technology, Banaras Hindu University, India in May 2001. He is a senior member of IEEE.