A Comparison on Hyperledger Consensus Mechanism Security and their Applications

Sara Nikolić[1,*], Nemanja Zdravković[1], Igor Franc[1] and N. Arivazhagan[2]

[1]Belgrade Metropolitan University, Faculty of Information Technology, Tadeuša Košćuška 63, Belgrade, Serbia
[2]SRM Institute of Science and Technology, Department of CI, Chennai, India

Abstract

In recent years, it has become evident that blockchain technologies are becoming more relevant, not just in the fintech industry, but rather in other fields as well. Whereas Bitcoin and Ethereum blockchains focus mostly on cryptocurrency, Hyperledger-based blockchains, or distributed ledger technologies (DLTs), focus on different use-cases, e.g. medicine, supply chain management, and public sector. Furthermore, Hyperledger-based DLTs can be viewed as a suite of frameworks, each designed to best fit into a certain purpose in terms of scalability, complexity, and security. Although similar, each DLT supports different consensus mechanisms, one if the building blocks of a blockchain network.

In this paper, we analyze the security properties of the most popular Hyperledger DLTs' consensus mechanisms, with an emphasis on security and computational complexity. We focus on the similarities and differences between the DLTs, comparing them to Bitcoin's proof-of-work and Ethereum's proof-of-stake, with the goal to find the most and least secure.

Keywords

blockchain, consensus mechanisms, distributed ledger technology, hyperledger, security

1. Introduction

Even though initially used in cryptocurrencies like Bitcoin and Ethereum, blockchain technologies (BCTs) have long surpassed this use-case [1, 2]. Transactions that form blocks which in turn form the chain are the key aspect of all BCTs. However, we can differentiate between several methods of block formation, level of access and, of course, consensus mechanism used. Hyperledger represents an opensource code that is under the Linux foundation. This collaboration initiative began as an idea to promote and improve BCTs and expand the possibility of its use in the business use-cases, for enterprise level solutions. By using Hyperledger BCT, which encompasses multiple distributed ledger technologies (DLTs), libraries and tools in the business processes, we can improve transparency, enhance liability and trust between business partners. However, for different BCT-based applications, depending on the need, some of the DLTs are more suitable than others [3].

The aim of this paper is to examine the most popular Hyperledger projects: Fabric, Sawtooth, Iroha, Indy, Burrow and Besu, in terms of security properties of the consensus mechanisms used, and to make the comparative analysis of the consensus mechanism vulnerabilities and possible solutions. In this review paper we wanted to emphasize the key characteristics and differences regarding the security aspects of consensus mechanism which are used in all of the hyperledger Blockchains, with the goal of answering the question of which consensus mechanism is the most secure.

This paper is organized as follows. After the Introduction, we examine the different variants of Hyperledger DLTs. Afterwards, we analyze the consensus mechanisms used in the DLTs. The results section focuses on vulnerabilities of each mechanism and possible solutions to overcome those vulnerabilities. Finally, we give some concluding remarks.

2. Hyperledger distributed ledgers

Blockchain technologies represent a dramatic improvement to the landscape of information collection, distribution, and governance [4]. Blockchain technology was originally the name given to the design that underpins the operation of the digital currency Bitcoin. Bitcoin's creator never used the term "blockchain" in his whitepaper, and reading it gives the distinct impression that the author was not introducing a new technology in the traditional sense, but rather a software design drawing on several existing technologies to allow him to create a "purely peer-to-peer version of electronic cash". The essence of Bitcoin's blockchain operation is that whenever two network members interact through a transaction, they announce their transaction to all network members (nodes), who record the transaction into a block with a limited capacity. Once the block is full, nodes perform the so-

BISEC'22: 13th International Conference on Business Information Security, December 03, 2022, Belgrade, Serbia
*Corresponding author.
sara.nikolic@metropolitan.ac.rs (S. Nikolić); nemanja.zdravkovic@metropolitan.ac.rs (N. Zdravković); igor.franc@metropolitan.ac.rs (I. Franc); arivazhn@srmist.edu.in (N. Arivazhagan)

called Proof-of-Work operations that are mathematically difficult to solve, although with predictable results, i.e. the correct solution is simple to verify. These mathematical operations have nothing to do with bitcoin transactions, but they are critical to the system's operation, because they force verifying nodes to expend processing power that would otherwise be wasted if they included any fraudulent or invalid transactions. The first node to correctly solve the Proof-of-Work problem broadcasts the solution to all other nodes, along with the block of transactions. Nodes can quickly and efficiently verify the accuracy of transactions and solutions, and when 51% of the network's processing power votes to approve a block, nodes start recording new transactions to a new, amended block [5].

Blockchain imposes fundamental changes to the way personal data are currently being processed, and can improve current data security solutions. A Blockchain is therefore a shared, append-only distributed ledger, in which all transactions, which can describe events (e.g. changes to bank accounts, updates to an electronic health record, each step in a supply chain, etc.) are stored in linked blocks [6].

The Hyperledger Project is a collaborative effort to develop an open-source, enterprise-grade distributed ledger framework and code base. It seeks to develop blockchain technology by finding and implementing a cross-industry open standard platform for distributed ledgers that has the potential to change the way commercial transactions are handled globally [7]. All Hyperledger projects are open source and free to use. Except for Hyperledger Indy which focuses on decentralized identity, all Hyperledger projects focus on general purpose applications. However, the consensus mechanism is one of the fundamental differences between the projects. Due to the wide range of blockchain usage needs, Hyperledger is developing a number of different consensus mechanisms [3]. Table 1 shows a summary of the key similarities and differences between Hyperledger projects [3]. The purpose of Hyperledger is to therefore provide efficient and robust blockchain systems and distributed ledgers for the creation of distributed services, i.e., systems in which all nodes in a network have the same copy of a ledger that can be read and edited independently by individual nodes. Hyperledger presently hosts sixteen projects, which are organized into four macrocategories: Distributed Ledgers, Libraries, Tools, and Domain-Specific. The six frameworks described below are the main strengths of Hyperledger [8].

2.1. Key Hyperledger Projects

Hyperledger Besu is an Ethereum client that is intended for enterprise use in both public and private permissioned networks. It can also be tested on Rinkeby, Ropsten, and Gorli test networks. Hyperledger Besu includes several consensus algorithms such as Proof-of-Work (PoW) and Proof-of-Assignment (PoA). Its comprehensive permissioning schemes are designed specifically for use in a consortium environment [8].

In terms of usage, potential, and efficiency, Hyperledger Fabric is one of Hyperledger's leading projects. Fabric's architecture is highly modular and configurable, allowing for innovation, versatility, and optimization across a wide range of use cases, including banking, finance, insurance, healthcare, human resources, supply chains, and even digital music distribution. Fabric is intended to serve as a foundation for developing modular applications or solutions. It allows components, such as consensus and membership services, to be plug and-play. It also provides a unique approach to consensus that enables performance at scale while maintaining privacy [8].

Hyperledger Indy focuses on Decentralized Digital Identities (DIDs) and intends to help those application domains where DIDs play an important and vital role. As a result, it offers tools, frameworks, and reusable components to make digital identities based on blockchains or other distributed ledgers compatible across administrative domains, apps, and any other silo. Indy is interoperable with various blockchains and may be used independently to power identity decentralization. With its unique consensus and ordering service algorithms, comprehensive role-based permission model, and multisignature support, Hyperledger Iroha is an easy-to-use modular distributed blockchain platform [8].

With its unique consensus and ordering service algorithms, comprehensive role-based permission model, and multi-signature support, Hyperledger Iroha is an easy-to-use modular distributed blockchain platform [8]. Hyperledger Iroha concentrates on the evolution of applications of mobile in combination with client libraries for both iOS and Android. This is a well-organized set of libraries and components. The synchronization and storage of data are performed off-device, and default network-wide repudiation system is done to verify validated nodes [9].

Hyperledger Sawtooth provides a flexible and modular architecture that separates the core system from the application domain, allowing smart contracts to express business rules for applications without needing to know the underlying core system design. It supports a variety of consensus algorithms, such as Practical Byzantine Fault Tolerance (PBFT) and Proof of Elapsed Time (POT) (PoET) [8]. Hyperledger Sawtooth is an open source distributed ledger designed for the modern enterprise. Unlike many popular blockchains, Sawtooth is not built for cryptocurrency, but instead for business supply chain management. To enhance performance, Sawtooth executes transactions in parallel instead of serially over a REST API. Sawtooth currently supports four alternative

Table 1
key similarities and differences between various Hyperledger projects [3].

Project	Sawtooth	Fabric	Indy	Burrow	Iroha	Besu
Advantages	Distributed state agreement, Adapters for transaction logic, Versatility, Scalability, Transaction families	Enterprise backing, Relative maturity, Private channels, Modular architecture, Smart contracts	Identity management	Lower barrier to entry, Use of the Ethereum Virtual Machine (EVM)	Mobile libraries	Network client variety, Plugins, Monitoring on Besu
Consensus mechanism	Proof of Elapsed Time, Practical Byzantine Fault Tolerance, Raft, Devmode	Kafka, Raft, Solo	Redundant Byzantine Fault Tolerance (RBFT)	Tendermint	Sumeragi	PoW, and PoA (IBFT, IBFT 2.0, Etherhash, and Clique)
Smart contract technology	Transaction Families	Chaincode	None	Smart contract application engine	Chaincode	
Smart contract type	On-chain and Installed	Installed	None	On-chain	On-chain	
Smart contract language	C++, Go, Java, JavaScript, Python, Rust, or Solidity (through Seth)	Go, Java, Javascript, Solidity	None	Native language code	Native language code	Java
State storage	Central lmdb database	CouchDB or leveldb	RocksDB	Google's Protocol Buffers	Kura	RocksDB

consensus algorithms. There really are four of them: Dev mode, PoET, PoET-Simulator, and RAFT [10].

Developers and architects can use Hyperledger Burrow to design an Ethereum virtual machine environment within the context of their Fabric and Sawtooth networks. It is a solution for combining Ethereum functionality with Hyperledger functionality [8]. Monax created Hyperledger Burrow, which Intel guarantees [9]. Burrow is a permissioned blockchain in which nodes, similar to the EVM, carry out smart contracts. Hyperledger Burrow is designed for multichain environments with application specifc contracts but coordinating a different domain.

Palma *et al.* [8] presented a summary utility tree of the most important technological decisions behind the architectural features of the Hyperledger frameworks implementations that may be able to cope with user requirements, as showed in Fig. 1. The symbol [*] means "every ledger" considered in [8].

3. Consensus mechanisms analysis

A consensus mechanism is a protocol in place to ensure that all participants in the blockchain network follow the agreed-upon rules [4, 11]. It assures that the transactions come from a legitimate source by requiring that

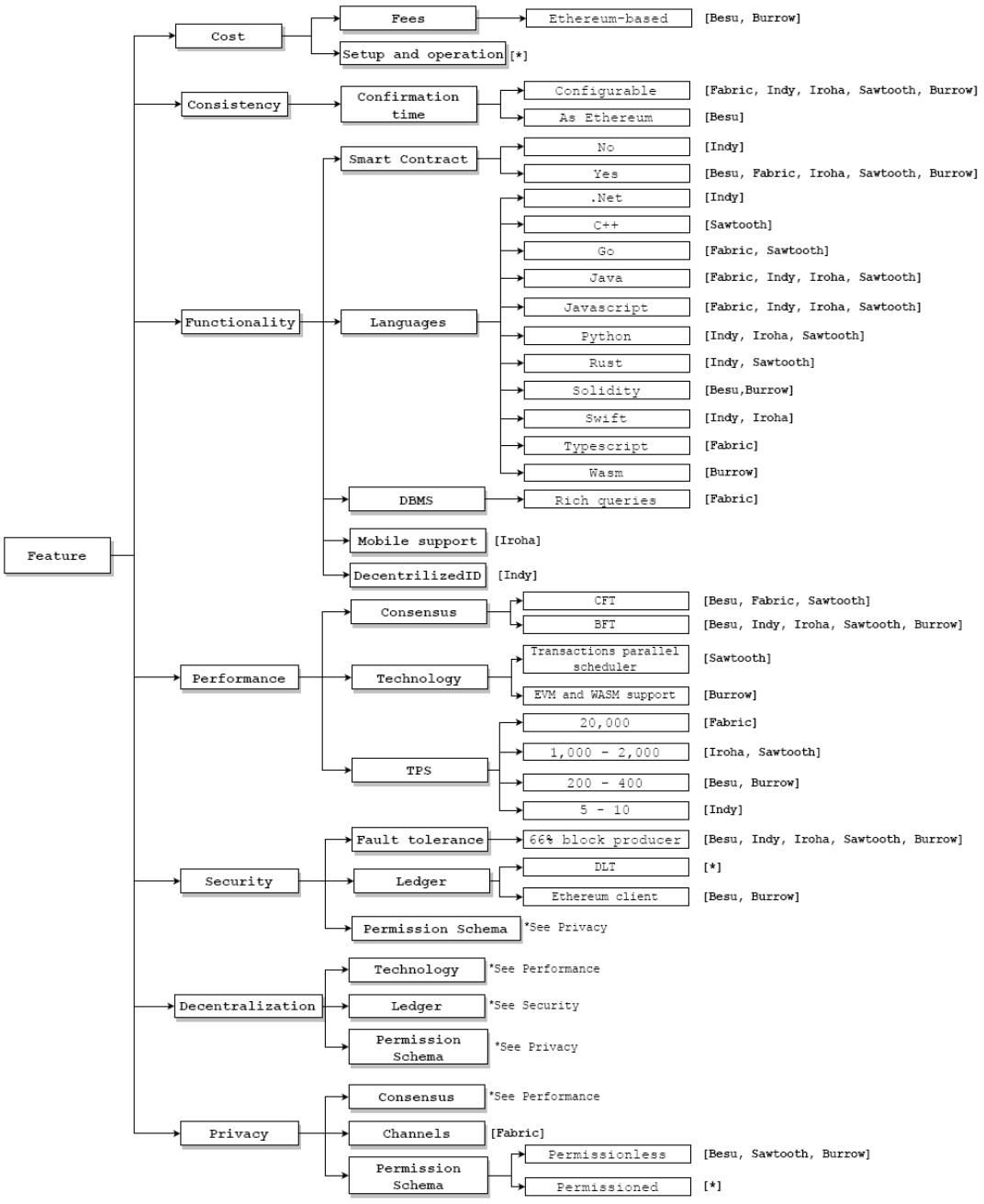

Figure 1: Summary utility tree of the most important technological decisions behind the architectural features of the Hyperledger frameworks implementations that may be able to cope with user requirements [8].

every participant agree to the status of the distributed ledger. The public blockchain is a decentralized technology, and there is no centralized authority to control the required act. As a result, the network requires authorizations from network participants for the verification and authentication of all blockchain network activity. The entire process is based on network participants consensus, which makes blockchain a trustless, secure, and

dependable platform for digital transactions [12]. There are several well-established methods by which different nodes in a blockchain network can reach consensus over a new block [13].

Proof of Elapsed Time (PoET) is an Intel-proposed consensus mechanism that operates similarly to PoW but consumes substantially less energy. Miners must solve a hash problem similar to PoWin this manner. Instead of a competition among miners to solve the next block, the winning miner is chosen at random based on a random wait time. The miner whose timer runs out first is the winner [13]. Its core mechanism is based on Intel's Software Guard Extensions (SGX) technology [14] that has the ability to digitally attest that some code has been correctly set up in aso called "Trusted Execution Environment" [15]. In PoET, this code is a function that generates a random time period that must be waited out by each node [16]. PoET is a scalable and efficient mechanism, particularly for permissioned networks. It generates a randomized model for picking block producers that does not require resource-intensive processing as in PoW systems or sophisticated calculations for deciding miners as in PoS and PoI consensus processes. PoET, on the other hand, is significantly reliant on Intel's specialized, third-party hardware to function, which increases entrance barriers for participants who do not have access to the SGX technology. It is also feasible that nodes with greater hardware available will have a better chance of being chosen, although such nodes will almost certainly be refused access to the permissioned network [16].

To solve the Byzantine generals problem, Practical Byzantine Fault Tolerance (PBFT) is applied. To work normally, PBFT can accept malicious behavior from up to one-third of all nodes. For example, in a system with one malicious node, at least four nodes are required to reach a proper consensus. Otherwise, consensus is not reached. When compared to proof of work, this method achieves consensus faster and more cheaply [13].

The Tendermint protocol is based on the PBFT consensus method. To select validators as participants, the protocol uses a voting method. Validators verify that adding blocks to the chain structure is done correctly. This ensures less number of nodes act as Validators and solves the computational complexity of PoW in energy-constrained environments [17]. A designated delegate will reach consensus in the RAFT protocol, and he will also be responsible for duplicating the logs whenever a new user joins the network. The heartbeat message will serve as an interrupt signal in this case, signalling the presence of the leader. If they do not receive the message before it expires, all nodes have a timeout mechanism in place. Then, unless the timer is reset, they will begin the process of electing a new leader. This protocol is more compatible with permissioned and private networks [17].

Proof of Work (PoW) was the first prominent blockchain mining mechanism presented in the literature used by the Bitcoin blockchain [1] and later it was adopted by the other cryptocurrencies like Litecoin, Ethereum, Monerocoin, and Dogecoin. It involves high algorithm cost with an open quorum structure [17]. The computationally expensive problem is crucial to the PoW mechanism: it must be difficult enough to solve to disincentivize attackers who want to contaminate the blockchain due to the high costs of obtaining a solution. Similarly, validating the suggested solution must be easy so that it may be easily accepted by other nodes and the correctness of the solution is transparent to the network, regardless of the computational power of any network node. Bitcoin's difficulty is to determine a value known as a "nonce." This nonce is generated by merging the proposed block's content to generate a new hash output that falls within a target range, such as a target hash prefixed with a number of 0's. The desirable output of a nonce can only be determined by brute force due to the nature of hashing algorithms. As a result, it is highly unexpected which node will successfully mine the next block, protecting validated transactions from tampering. PoW has effectively sustained and protected the operations of two of the most popular public blockchains, Bitcoin and Ethereum, by requiring expensive computational power and information transparency [16]. By establishing a large number of malicious nodes, the Sybil attack can successfully exploit the PoW. The balancing attack can be used against the Ethereum protocol and private blockchains. Furthermore, DDoS attacks and BGP hijacking can be used to interrupt the regular flow of this consensus mechanism [12].

Proof of Authority (PoA) is designed to optimize the PoS mechanism and be used in permissioned networks. Instead of selecting block miners based on their stakes in cryptocurrency tokens, PoA selects a small group of authorities as transaction validators based on their network identification or reputation [18]. A PoA-based system also rewards authorities for certifying and ordering transactions to incentivize honest behavior in providing service and moderating the network [16]. PoA does not necessitate intensive processing to execute difficult tasks and depends on a limited number of validators to obtain consensus. When compared to PoW and PoS-based systems, these features help improve transaction throughput and energy efficiency. PoA, on the other hand, avoids decentralization by concentrating mining power among a handful of trusted authorities. As a result, this paradigm has the potential to introduce censorship into the public network, with one or more authorities blacklisting or denying all transactions from a specific user. On the other hand, a permissioned network built between multiple businesses or major institutions might profit from PoA since it offers a faster transaction processing speed and the identity-at-stake model aligns well

with business operations that value trustworthiness and reputation [16].

Apache Kafka is a distributed streaming platform, based on a commit log that started as an internal LinkedIn project designed to provide a low-latency, high-throughput platform for manipulating real-time data feeds [19]. The typical Bitcoin system takes roughly 10 minutes to build blocks. As a result, we simplified the standard blockchain's consensus algorithm and used Kafka distributed message processing instead of PoW, PoS, and other consensus mechanisms to provide record verification and backup on the block alone. Kafka consensus selects several fixed nodes to implement the Kafka cluster to maintain partition logs, and the remaining nodes are used as transaction production and consumers to manage messages in the queue. The Kafka consensus has the advantage of having high throughput and low latency. The disadvantage is that it can only tolerate 1/2 of the maximum node failure, and cannot tolerate the existence of malicious nodes [20].

Sumeragi is the name of the Byzantine fault tolerant distributed consensus algorithm used by Hyperledger Iroha. Most of the algorithm is based on the B-Chain consensus algorithm. In Sumeragi, consensus is performed on individual transactions and the global state resulting from all transactions [21].

4. Security properties of consensus mechanisms

The well know cryptocurrency, Bitcoin, utilizes a permissionless public blockchain framework. Being permissionless, it allows any node to participate in the consensus protocol and mine blocks without any permission. It uses PoW as the method for consensus which has a high latency about 10 minutes, making it ineffective for IoT networks. However, it is worth exploring if it can still be used with eased proof of work to reach a consensus in a short time [13]. Bitcoin's PoW mechanism consumes massive power. It is estimated that power consumption of Bitcoin per transaction is around 545 KWh [22].

Ethereum is a permissionless public blockchain framework developed using solidity which is a contractoriented, high-level language for implementing smart contracts . All the nodes are required to participate in the consensus process. This method is significantly less computational-intensive than the original PoW. This blockchain can be customized and adopted for a variety of applications because of its intrinsic characteristics that enable smart contracts. Its block generation process takes between 10 to 20 seconds which is much less than bitcoin's latency [13]. It is estimated that power consumption of Ethereum per transaction is around 49 KWh [22].

The most common vulnerabilities and attacks on BCTs and DLTs based on consensus are DDoS attacks, 51% attacks, long range attacks, $P + \epsilon$ attacks, and Sybil attacks. DDoS attacks on a blockchain focus on the protocol layer, with the biggest threat to blockchains being transaction flooding, when spam and false transactions flood the blockchain. The attacker can hence compromise the availability for original users and undesirable have other impacts on the network. The vulnerable mechanisms are PoW, Delegated PoS, RAF and PoA. A possible solution for these attacks are transaction filtering.

The next vulnerability is the 51% attack. The 51% attack is committed when a miner or group of miners gains control of more than 50% of a network's blockchain. This threat is targeted to cryptocurrencies, and in most situations, it cannot be detectable until it's too late. The vulnerable mechanisms are RAFT, PoW, PoS, and delegated PoS. Although a possible solution for this attack does not exist, the larger the network, this type of attack is less likely to happen.

Similar to the 51% attack, long range attacks can occur when a false chain takes over the correct chain, rendering the previous chain invalid. The only mechanism that is vulnerable to this type of attack is PoS due to small intervals of block generation. Possible solutions include bootstrap nodes, checkpoint, or adding a range a blocks to always be considered true.

The next type of attack is the so-called $P + \epsilon$ attack, where a malicious node, i.e. the attacker, is influencing other nodes that get incentives from the blockchain. The attacker promises to pay $P + \epsilon$ to all voters who vote for attacker's option, if the majority votes for attacker's option. Distributed PoS and PoW mechanisms are vulnerable to this attack; however, this type of attack is very hard to implement.

Finally, Sybil attacks occur when a single node makes many fake identities (called Sybil identities) simultaneously. All mechanisms are theoretically vulnerable to Sybil attacks. More effort on authentication, such as Two-Factor Authentication (2FA) or node propagation modeling could be possible solutions to these attacks.

A summary of the vulnerabilities is given in Table 2.

5. Conclusion

In this paper, we have analyzed the most common Hyperledger project DLTs used, with emphasis on their consensus mechanisms used and the security properties of each. While some mechanisms are more resilient than others, all consensus mechanisms is at least vulnerable to at least one kind of threat or attack. As far as attacks are concerned, the majority of them are focused on cryptocurrency BCTs, while enterprise DLTs are less likely to be a target of an attack.

Table 2
Types of most common attacks/vulnerabilities on BCTs/DLTs based on consensus.

Vulnerability/Attack	Vulnerable mechanisms	Possible solution
DDoS blockchain attacks	PoW, PoS, DPoS, RAFT, PoA	Transaction filtering
51% attacks	RAFT, PoW, PoS, DPoS	Hard to implement the attack
Long range attakcs	PoS	Bootstrap nodes, Checkpoints, Range of blocks to always be considered true
$P + \epsilon$ attacks	PoW, DPoS	None; Hard to implement the attack
Sybil attacks	All mechanisms	2FA, Node propagation modeling

Currently, all BCTs/DLTs use one form of consensus mechanism, and a possible solution for added security could be to combine two or more consensus mechanisms for a future DLT implementation. Furthermore, policies such as node segregation, node addition "cost", and node behavior monitoring can be implemented to secure public BCTs. We can conclude that the use of private blockchains/distributed ledgers is so far a better choice in terms of security, and future work will focus on improving existing mechanisms.

Acknowledgment

This paper was supported by the Blockchain Technology Laboratory at Belgrade Metropolitan University, Belgrade, Serbia.

References

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized business review (2008) 21260.

[2] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32.

[3] V. Milicevic, J. Jovic, N. Zdravkovic, An overview of hyperledger blockchain technologies and their uses, in: ICIST 2021 Proceedings, 2021, pp. 62–65.

[4] J. J. Bambara, P. R. Allen, Blockchain, A practical guide to developing business, law and technology solutions. New York City: McGraw-Hill Professional (2018).

[5] S. Ammous, Blockchain technology: What is it good for?, Available at SSRN 2832751 (2016).

[6] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in: 2017 IEEE international congress on big data (BigData congress), Ieee, 2017, pp. 557–564.

[7] C. Cachin, et al., Architecture of the hyperledger blockchain fabric, in: Workshop on distributed cryptocurrencies and consensus ledgers, volume 310, Chicago, IL, 2016, pp. 1–4.

[8] S. Dalla Palma, R. Pareschi, F. Zappone, What is your distributed (hyper) ledger?, in: 2021 IEEE/ACM 4th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), IEEE, 2021, pp. 27–33.

[9] R. L. Kumar, Y. Wang, T. Poongodi, A. L. Imoize, Internet of Things, Artificial Intelligence and Blockchain Technology, Springer, 2021.

[10] B. Ampel, M. Patton, H. Chen, Performance modeling of hyperledger sawtooth blockchain, in: 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE, 2019, pp. 59–61.

[11] A. Baliga, Understanding blockchain consensus models, Persistent 4 (2017) 14.

[12] S. Sayeed, H. Marco-Gisbert, Assessing blockchain consensus and security mechanisms against the 51% attack, Applied sciences 9 (2019) 1788.

[13] M. Salimitari, M. Chatterjee, An overview of blockchain and consensus protocols for IoT networks, arXiv preprint arXiv:1809.05613 (2018) 1–12.

[14] F. McKeen, I. Alexandrovich, I. Anati, D. Caspi, S. Johnson, R. Leslie-Hurd, C. Rozas, 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, 2016, pp. 1–9.

[15] M. Sabt, M. Achemlal, A. Bouabdallah, Trusted execution environment: what it is, and what it is not, in: 2015 IEEE Trustcom/BigDataSE/Ispa, volume 1, IEEE, 2015, pp. 57–64.

[16] P. Zhang, D. C. Schmidt, J. White, A. Dubey, Consensus mechanisms and information security technologies, Advances in Computers 115 (2019) 181–209.

[17] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, W.-C. Hong, A survey on decentralized consensus mechanisms for cyber physical systems, IEEE Access 8 (2020) 54371–54401.

[18] G. Wood, Poa private chains, 2015. URL: https://github.com/ethereum/guide/blob/master/poa.md.

[19] M. Petrescu, R. Petrescu, Log replication in raft vs kafka, Studia Universitatis Babeș-Bolyai Informatica 65 (2020) 10–24193.

[20] G. Luo, M. Shi, C. Zhao, Z. Shi, Hash-chain-based cross-regional safety authentication for space-air-ground integrated vanets, Applied Sciences 10 (2020) 4206.

[21] Z. Hintzman, Comparing blockchain implementations, SCTE-ISBE Cable-Tec EXPO (2017).

[22] N. Chaudhry, M. M. Yousaf, Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities, in: 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), IEEE, 2018, pp. 54–63.