Development of a Linear-Scaling Consensus Mechanism of the Distributed Data Ledger Technology

Gennady Shvachych, Ivan Pobochii, Hanna Sashchuk, Oleksandr Dzhus, Olena Khylko, and Volodymyr Busygin

Abstract The paper proposes and explores a new blockchain system that operates on a linearly scalable consensus mechanism. This selection method confirms the shard through shares voting and scalable random generation by VDF (Verifiable Delay Function) and VRF (Verifiable Random Function). The system analyzes available consensus mechanisms, sharding, and the age of distributed randomness. It is energy efficient, fully scalable, secure, with fast consensus. Compared to available methods, the improved shard method performs network connection and transaction verification and reveals the state of the blockchain. The threshold has a sufficiently low coefficient for small validators to participate in the network and receive rewards. The proposed sharding process runs securely due to a distributed randomness (DRG) process that is unpredictable, impartial, and verified. The network is constantly overloaded to prevent slow adaptive Byzantine malicious validators. Contrary to other sharding blockchains that require Proof-of-Work to select validators, the proposed consensus is attributed to Proof-of-Stake, therefore, energy-efficient. Herein the consensus is achieved by a BFT algorithm which is linearly scalable and faster than PBFT.

1 Introduction

A distinctive feature of the innovative technology of the distributed data ledger (blockchain) presented in the form of mathematical algorithms and software is that it requires no participation of contractors when concluding contracts allowing transactions to be made without intermediaries such as enterprise banks and lawyers.

Literature analysis shows that the practical application of blockchain technology does not have in-depth subject coverage. Blockchain is mainly seen as a general purpose technology. Paper [1] provides specific examples of companies using blockchain. Moreover, it highlights that blockchain publications are usually predictive, highlighting the potential of the technology extensively, but there is no discussion yet about how blockchain can improve enterprise efficiency. In the publications under review, the main focus is on what can happen if the blockchain is massively implemented in enterprises. The paper also highlights the lack of research that details the implications of blockchain applications for entrepreneurs and describes their entrepreneurial aspects. Similar views are shared by other researchers [2, 3]. On the other hand, analysis of the literature review shows that the competitiveness of blockchain technology is reflected through the choice of technology. It was revealed that approaches to applying blockchain technology could be implemented according to two central schemes: “technology first—then a problem” or “first a problem—then a technology.” However, studies have shown that enterprises with the extensive implementation of blockchain technology tools tend to operate by the latter scheme. Hence, the problem is considered, followed by justifying the problem’s solution through the blockchain. Researchers note that this is the most effective approach [4].

Analysis of methods for implementing blockchain technology based on the capabilities of the already created Ethereum and Bitcoin blockchains has shown certain drawbacks for their use in any area, including in the digital economy. At the same time, some methods of blockchain technology require improvement. For instance, the innovative Bitcoin blockchain was meant to become a peer-to-peer payment system allowing transfer funds and excluding intermediaries such as payment systems or banks. However, Bitcoin gained some shortcomings for its limited bandwidth—around seven transactions a second, which became pretty expensive as a payment system. Soon, a new blockchain infrastructure, *Ethereum* [5], allowed developers to develop different types of blockchain applications via smart contracts. Nevertheless, Ethereum, with 15 transactions a second, was unable to help high-performance applications such as games or decentralized exchanges and did not solve the scalability problem. Given the performance limitations of Ethereum and Bitcoin, several blockchain projects offered different solutions trying to boost transaction output.

Other blockchains proposed replacing the Proof-of-Work consensus with Proofof-Stake. Various blockchains, e.g., EOS, apply Delegated-Proof-of-Stake (DPoS) consensus, where a vote elects the chain of blocks rather than through the process of chain algorithmic. Several chains like IOTA [24] changed the blockchain structure of data by a Directed Acyclic Graph (DAG), disrupting transactions' interconnected post-processing. Nevertheless, those solutions cannot significantly increase performance [6] without sacrificing other essential aspects such as decentralization and security [7, 8]. It becomes obvious that the most valuable link in blockchain technology is the algorithms for reaching consensus because those provide it with reliability. Research data aim at consensus mechanism further development of the distributed data ledger technology.

2 Analysis of Recent Research and Publications

The consensus protocol is a crucial factor of a blockchain that determines the level of security and speed of blockchain validation to reach a consensus for the next block. The *Proof-of-Work* was the first blockchain consensus protocol provided by Bitcoin. *PoW* means that when the miner solves a cryptographic puzzle, and if succeeded could offer the next block and receive symbolic rewards. Honest nodes control over 50% of the hashing power. The consensus rule herein means that the longest chain remains the only correct; therefore, *PoW* consensus is based on the chain. Such a consensus has the main drawback: if someone has at least 1% more capacity than the rest of the network, i.e., 51% or more, a kind of "controlling stake" of generating capacity; in this case, one can single-handedly control all operations via the system, create blocks, confirm or block transactions. Note that a hash in such a protocol is a set of 64 alphabetic and numeric characters, and the complexity regulates the number of zeros at its beginning. For instance, we have the following hash 0000045c5e2b3911eb937d9d8c574f09. The main proof-of work process is mining. It consists of iterating over a numeric value until the block header looks like it should. After all the necessary conditions are met, the miner publishes a block indicating all the necessary attributes, including the found value. Knowing all the attributes, the complete ones automatically check whether the header hash will look exactly like this and not otherwise with such initial data and such a found value. After confirmation, the miner switches to generating a new block, and the author of the newly created block receives a reward on own Bitcoin wallet [9]. In the *Proof-of-Stake* approach, nodes also try to hash data, searching for a specific value result. However, the complexity is distributed proportionally and in compliance with the node's balance according to the number of coins (tokens) in the user's account. Thus, there is a better chance of generating the next block node with a larger balance. Unlike *Proof-of-Stake*, the algorithm spends much less power. Another kind of consensus protocol is represented by *PBFT* (*Practical Byzantine Fault Tolerance*) (Fig. 1).

Named after the mathematical puzzle of Byzantine Generals Problem [10], when several Byzantine generals surrounded the city with their armies, they must agree on actions when attacking or retreating. If the generals do not agree upon the decision, the operation leads to disaster. One "leader" node and other "validators" nodes in *PBFT*. Each *PBFT* consensus round includes two main stages: the Prepare and the Commit stage. During Prepare stage, a leader passes on their offer to every validator, who give votes for the request to all the others. The re-relaying is essential for all validators as the rest of the validators must count the votes of each validator. The preparatory stage ends when more than $2f + 1$ observe consecutive voices when f is the quantity of malicious validators and the absolute quantity of validators plus one, the leader $3f + 1$. The commit stage covers an akin computing process, with the reached consensus when $2f + 1$ sequential voices are observed through relaying votes between validators of *PBFT* $O(N)2$

complexity of communication, nonscalable for a blockchain system with a huge number of nodes.

Fig. 1 Graphical interpretation of the PBFT consensus protocol. *Source* Authors’ elaboration

So, Fig. 1 shows that the *PBFT* protocol is a type of Byzantine state machine system that requires the state to be kept together and all nodes to perform the same actions. To do this, three types of base agreements must be met, including an agreement, a review agreement (expiration period), and a view change agreement. Obviously, with this, the *PBFT* protocol mechanism does not require mining or huge computations, so it takes a short time to reach a consensus. Currently, the *PBFT* protocol is flexibly improved and combined, e.g., with the *PoW* or *PoS* algorithms. Thus, new, hybrid consensus mechanisms are formed. Let us consider one of them. Using *PBFT* and *PoW* hybrid consensus algorithms, the former generates fast chains while the latter generates slow ones. Transaction confirmation and mining are separated. After packaging the transaction, the *PBFT* committee confirms its creation by *fastBlock*. The transaction is approved; the slow chain will pack the *fastBlock* in the fast chain into a *snailBlock*, which the miner validates to reach the chain. With this hybrid consensus algorithm, *tps* has been greatly improved, and the application of *PoW* mining implemented the decentralization idea. The *PBFT* committee changes every two days, and all candidate members become miners after successfully mining *PoW*, which ensures the principle of honesty and fairness. Although the *PBFT* algorithm was originally designed to serve both private and public networks, it continues to improve and flexibly modify. Such a protocol shows that the public network will play an important role in the consensus mechanism in the future. In this regard, this paper presents one of the options for its improvement.

Research objectives. Based on the literature review and the analysis of the current development of blockchain technology problems, develop a fully scalable, evidence-based secure, and energy-efficient blockchain; explore the functionality and features of a blockchain system based on sharding the next-generation solving several problems of available blockchains.

3 Main Research Material Statement

3.1 Research and Analysis of Scalability and Security Mechanism and Decentralization of the Blockchain Technology

The solution to the scalability providing security and decentralization simultaneously is sharding that builds groups of validators allowing transactions to be processed simultaneously. Thus, the entire output of transactions increases with the quantity of participants linearly. The *Zilliqa* blockchain became the first public blockchain to offer a solution to the problem of scalability with sharding. However, the blockchain fails to meet two fundamental requirements. First, it shares no data storage with the blockchain (shorted state). That prevents local computers from participating in the network, thereby limiting decentralization. Second, the *PoW* based blockchain consensus algorithms consume massive computational resources. Sometimes, users cannot get powerful computing power in many scenarios, and all mining-based consensus algorithms face low transaction speed. Solving the blockchain system scalability problem restricts the technology application in various areas of the digital economy. Some developers are proposing a parallel distributed structure of a distributed cloud storage system and a decentralized system for blockchain to solve scalability, large-scale storage, and data exchange. The proposed method demonstrates a new hybrid consensus protocol for large-scale public blockchain based on joint optimization. Prevention of the Sybil attack is a crucial security factor in public blockchains. A Sybil attack is a peer-to-peer attack that only connects the victim to nodes controlled by the attacker. In peer-to-peer networks, where no host is trusted, each request is duplicated for multiple recipients so that no single host can be fully charged. Meanwhile, network users can have multiple identifiers that are physically associated with different nodes. Those identifiers can share resources or have multiple copies of them. The latter will create a backup that will check the integrity of the data taken from the network on its own. The downside is that at some point, more sites that are supposed to represent different recipients of a particular request can be

controlled by the same user. Moreover, consider the user becomes an intruder. In that case, the latter will have all the capabilities of an intermediary at this session and unjustifiably get the complete trust of the session initiator. The more identifiers an attacker has, the more likely the next user session will be closed. Figure 2 presents the blockchain splitting into two chains as malicious nodes want to create blocks that do not correspond to the consensus. It is vital for an attacker that a new identifier is light enough to make [11]. *Bitcoin* and *Etherium* demand that miners solve a cryptographic puzzle ahead of offering a block. Furthermore, sharding blockchains such as *Zilliqa* [12] and *Quarkchain* [13] apply *PoW* to avoid Sybil attacks. Sharding is defined as dividing.

Fig. 2 Malicious nodes of the blockchain dividing it into two chains. *Source* Authors' elaboration and storing a single logical set of data in multiple databases. Sometimes sharding is recognized as horizontal data partitioning.

The sharding involves dividing the blockchain into separate segments (shards). A single shard contains a unique set of smart contracts and account balances. Each shard is assigned a node, identified transactions, and operations, in contrast to the method where each node accounts for verifying each transaction across the entire network. Dividing blockchain into more manageable segments can increase transaction capacity and solve the problem of scalability that most modern blockchains face (Fig. 3).

There are two shards in this blockchain (Fig. 3); both forks precisely when the transaction gets in block A of shard #1 and block X of shard #2. The shard must discard one chain and accept another one for the fork. Therefore, if shard #1 acquires chain A, B, and so on, and shard #2 acquires chain W, X, and so on, the consensus gets confirmation. If shard #1 earns chain A, B, and so on, and shard #1 is chain W, X, and so forth, the consensus is rejected and can be resent. If shard #1 acquires chain A, B, etc., and shard #2 acquires chain W, X, etc., one part of the transaction gets confirmation (A, B, etc.), while the other does not (W, X, etc.). Various sharding solutions are offered in both industry and science. *Zilliqua* was the first public blockchain based on sharding to claim an output of 2800 transactions a second in the industry. *Zilliqa* prevents a *Sybil* attack by applying *PoW* as a face registration process. The *Zilliqa* network uses the separate directory maintenance committee and network sharding, counting hundreds of nodes each. The transactions are processed solely and are appropriated to various shards. The accepted blocks from all shards get accumulated and combined in the maintenance committee directory. In academia, publications such as *Omniledger* [14] and *RapidChain* [15, 16] offer solutions where each sharding contains a subset of blockchain states. *Omniledger* uses *RandHound* [10], a multilateral computational scheme to generate a secure random number for nodes allocation to shards. *Omniledger* assumes an adaptive model, where case attackers can damage more and more nodes over time. According to this security model, one fragment can eventually be damaged. The *Omniledger* prevents damage to the shards by rearranging all nodes at a specified interval, an epoch (stage). *RapidChain* builds off the *Omniledger* and suggests a constraint rule to swap nodes without interruption [17].

Fig. 3 The process of a shard formation in the *Zilliqua* blockchain. *Source* Authors' elaboration

3.2 Study of the Blockchain Nodes Distribution in the Shard

At the moment, various approaches were proposed for the nodes' distribution in a shard: distribution based on randomness, location, and centralized control. The sharding based on randomness was found to be the most reliable solution of all the approaches. The sharding based

on randomness uses a mutually consistent random number for each node. Thus, a random number should cover the next features:
1. Erratic: nobody must foresee a random number.
2. Biased: random number generation ought not be tendentious by any member.
3. Checked: Any observer must check the generated random number validity.
4. Scalable: randomness generation algorithmmust scale tomasses of participants.
The *Omniledger* blockchain uses the *RandHound* protocol, driven by a leader distributed random case generation covering the Byzantine convention and *PVSS* (*Public Verified Secret Sharing*). *RandHound* is a protocol that distributes member nodes toward size groups. That completes the first three properties described above but slowly qualifies as scalable. While *RapidChain* takes a more straightforward approach, allowing each member to make *Verifiable Secret Sharing* (*VSS*), applying the combined, secret exchanges as the resulting randomness. However, since malicious nodes can transmit incompatible shared resources to different nodes, this protocol is not secure. Furthermore, *RapidChain* does not demonstrate how the nodes reach consensus on versatile versions of randomness. *Ethereum 2.0* proposes the delay check function to avoid a hacker attack by disclosing the actual random number. The delay function check is a cryptographic primitive; it takes a minimum adjustable time to calculate and check the result at once.

3.3 Principal Features of the New, Developed Blockchain System

The paper investigates the functionality and features of a blockchain system based on sharding the next generation, solving several blockchain problems to create a fully scalable, evidence-based, energy-efficient, and secure blockchain. The developed approach aims at improving available methods. It is notable that the fundamental differences between the proposed approach and available ones. The paper presents and investigates a distributed ledger system that runs on a linearly scalable consensus mechanism. This selection method confirms a shard by voting shares and has scalable randomness generation by *VRF* and *VDF* functions. Such a system is based on analyzing available consensus mechanisms, sharding, and distributed random generation. The proposed approach allows blockchain development with the following advantages: full scalability, security, energy efficiency, and fast consensus. Due to scalability and energy efficiency, the proposed method is suitable for creating a blockchain for the digital economy.

3.4 Development of a New Scalable Blockchain Consensus Protocol

As an improvement of the *PBFT* protocol, the thesis proposed a consensus mechanism scalable in a linear fashion regarding communication intricacy. Instead of inviting everyone to post votes, a leader starts signing a multi-signature to compile validator votes for*O*(1)multi-signature followed by relaying it. And in toward getting *O*(N) of signatures, a validator gets only one multi-signature, thereby cutting the communication complexity with *O*(N)2 to *O*(N). The multi-signature *O*(1) sense is a *BFT* method improvement from the *ByzCoin* blockchain [18] with the *Schnorr* signature scheme to aggregate consistent multivalued signals, creating the multicasting tree between the validators expedite message delivery. Nevertheless, *Schnorr's* multi-valued signature demands a secret series of commitments, resulting in two round-trip requests for a single multi-signature. The proposed method upgrades available one as the *BLS* (*Boneh-Lynn-Shacham*) multi-signature with only one round-trip request. Hence, the developed method is at least 50% faster than the *BFT ByzCoin* method. Figure 4 depicts the network communication of the developed process during one round of consensus. The developed method for conducting the consensus procedure covers the next steps:
1. The leader builds a new block and passes the block header to each validator. At the same time, the leader relays the block's contentswith the abrasion-encoding. The "declare" stage (Fig. 4—*Announce* stage).
2. The validators analyze the block header's validity, sign it by *BLS* signature, followed by relaying the signature back to the leader (Fig. 4—*Prepare* stage).

3. The leader awaits minimum of 2 *f* + 1 valid signatures from validators and merges them within the *BLS* multi-signature. Then a leader relays the aggregated multi-signature with the bitmap and changes signed by the validators. Along with step 3, the *PBFT Prepare* stage is completed.
4. Validators verify whether multi-signature includes minimum 2 *f* + 1 signers, verifying transactions with the block transmitted content from the leader via step 1, signs the message from step 3, and returns the message to the leader.
5. The leader awaits minimum 2 *f* + 1 valid signatures and, starting from step 4, combines them within a *BLS* multi-signature with a bitmap logging of everyone who signed. Then, the leader makes a new block including all signed multisignatures and bitmaps, followed by relaying a new block to each validator. Along with step 5, the *PBFT Commit* stage is completed (Fig. 4—the *Commit* stage).

Fig. 4 Network communication of the developed method during one round of consensus. *Source* Authors' elaboration

Proof-of-Stake selects consensus validators. The proposed protocol is different from available *PBFT* in that a validator that keeps massive voting shares has more votes than the others, instead of a single vote (signature). Contrary to waiting for minimum 2*f* + 1 signatures from validators. Further, the leader awaits signatures from validators with minimum 2 *f* + 1 voting shares. Note that the traditional procedure for downloading the history of blockchains and rebuilding the available state is too sluggish to allow re-making changes (it takes several days for the Ethereum blockchain to synchronize the history fully). The current state is much lesser than entire history of blockchain. Loading the present state across the epoch is possible compared to loading the entire history. To optimize the state synchronization, it is proposed to make the state of blockchain as the smallest. *Ethereum* has many empty accounts and wastes state-space on the blockchain. Empty accounts cannot be deleted due to possible replay errors when old transactions are re-sent to a deleted account. The problem can be solved by preventing replay attacks by allowing transactions to designate the current block's hash: the transaction is only valid up to a specific number of blocks after the specified hash's block. Hence, old accounts can be deleted, significantly speeding up the analyzing current blockchain state. Thus, new validators that attach to the shard first load the present shard state to quickly validate the transactions. The new node must perform an appropriate check to provide that the present loaded state is valid. Contrary to downloading the entire history of blockchain and re-making every transaction analyzing the present state, the new node downloads the initial block headers and verifies the headers by verifying the signatures. The state is valid for cryptographic follow from the present state to the initial block. Signature verification is not computationally hard, and it could take much time to verify all signatures, starting with the genesis block. The first block of each epoch is proposed to incorporate an additional hash pointer to the last epoch first block to mitigate the problem. Hence, a new node can traverse the blocks during the epoch as it archives hash-pointers track to the genesis block. That substantially boosts the verification of the present state of the blockchain. So, we note some features of the proposed approach. This paper presents and investigates a new blockchain system that operates on a linearly scalable consensus mechanism. This selectionmethod confirms a shard by voting shares and has scalable randomness generation by the *VRF* and *VDF* functions. The new system is based on an analysis of available consensus mechanisms, sharding, and distributed randomness generation.

The shortcomings analysis of available blockchain systems showed that the proposed sharding method performs network connection and transaction verification and reveals the state of the blockchain. The proposed consensus mechanism showed that the decision threshold is low enough for small validators to participate in the network and earn rewards. The sharding process covered in this paper is secure through the use of distributed randomness (DRG), which is unpredictable, unbiased, and proven. The network is overloaded continuously to prevent slow adaptive byzantine pests. Unlike other blockchains based on sharding and requiring a *PoW*-type transaction validation and confirmation model to select validators, the proposed consensus is based on applying a *PoS* model and, therefore, is more energy efficient. Consensus is reached by a linearly scalable *BFT* algorithm, which is more of a *PBFT*. Introducing protocols and network innovations gets a scalable and secure new blockchain system.

4 Conclusions

The paper proposes and explores a new blockchain system that operates on a linearly scalable consensus mechanism. This selection method confirms the shard by stock voting and has scalable random generation using the *VDF* (*Verifiable Delay Function*) and the *VRF* (*Verifiable Random Function*). The new system analyzes available consensus mechanisms, sharding, and generation of the distributed randomness. The proposed approach allows the development of a blockchain with the following advantages: full scalability, security, energy efficiency, and fast consensus. The shortcomings' analysis of available blockchain systems showed that the proposed sharding method performs network connection and transaction verification and reveals the blockchain state. The proposed consensus mechanism showed that the acceptance threshold has a sufficiently low coefficient for small validators to participate in the network and receive rewards. The sharding process covered in this paper is safe due to the distributed randomness (*DRG*). The *DRG* is changeable, impartial, and verified. The network is constantly overloaded to prevent slow adaptive Byzantine malicious validators. Unlike other blockchains built around sharding and require a *PoW*-type transaction verification and confirmation model to select validators, the proposed consensus is rooted in the *PoS* model and, therefore, more energy-efficient. Hence the consensus is achieved via a scalable *BFT* algorithm in a linear fashion, which is more of a *PBFT*. A scalable and secure new blockchain system is obtained by introducing innovations at the protocol and network levels. The methods for creating the blockchain improve available mechanisms with practical value for use in various digital economy sectors. Separately, we note some promising areas for the practical implementation of the research. Firstly, the considered technologies in supply chains are of undoubted interest. Logistics at the present development stage is one of the biggest problems for the current generation of companies. The industry is looking for new technologies to improve available processes, reduce costs, and increase transparency in the supply chain. That is where blockchain technology offers a solution to most current problems. Secondly, we highlight the proposed approach for implementation in the banking sector. This technology can completely transform the banks' structure, and soon it will become radically different from what we are used to today. Avoiding the mediation of third parties in various transactions can make a huge layer of banking services useless.

Thirdly, to date, a certain successful experience has been accumulated in blockchain solutions applications to ensure the integrity and authenticity of documents, information, and control information. In this regard, this direction is promising in terms of the implementation of the proposed research. The paper research demonstrates that one of the main problems of the studied technologies is in the features of the modeling process, both machine and mathematical [19–22]. For instance, for servicing and solving security problems of those technologies, it is necessary to use powerful computing equipment and high-performance ones. On the other hand, the issue of developing new blockchain systems, e.g., based on a linearly scalable consensus mechanism, can be solved only via up-to-date and complex mathematical apparatus. The authors attribute those problems to theprospect of further research on this topic.

References

1. M. Bjørnstad, J.Harkestad, S.Krogh, *What are Blockchain Applications?Use Cases and Industries Utilizing Blockchain Technology*. NTNU (2017). https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2472245/17527_FULLTEXT.pdf?sequence=1&isAllowed=y
2. S. Alvarez, L. Busenitz, The entrepreneurship of resource-based theory. J. Manag. **27**(6), 755–775 (2001)

3. S. Davidson, *Economics of Blockchain* (2016). SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2744751
4. G. Khachatryan, *Blockstack: A New Internet for Decentralized Apps - Grigor Khachatryan. Medium* (2018). https://grigorkh.medium.com/blockstack-a-new-internet-for-decentralizedapps-1f21cb9179b9#:%7E:text=Blockstack%20is%20a%20new%20decentralized,a%20force%20of%20monumental%20change
5. W. Mougayar, V. Buterin, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology* (1st ed., Wiley) (2016)
6. J.T. Kruthik, K. Ramakrishnan, R. Sunitha, B. Prasad Honnavalli, Security model for Internet of Things based on Blockchain, in *Innovative Data Communication Technologies and Application* (Springer, Singapore, pp. 543–557) (2021)
7. G. Danezis, S. Meiklejohn, Centrally banked cryptocurrencies, in *23rd Annual Network and Distributed System Security Symposium, NDSS*, 21–24 (2016)
8. R. Ivanov, V. Busygin, Some aspects of innovative blockchain technology application, in *Proceedings of the XII International Scientific and Practical Conference. Modern problems of modeling of socio-economic systems,* Kharkiv, Ukraine (2020)
9. B. Awerbuch, C. Scheideler, Towards a scalable and robust DHT, in *Eighteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '06* (pp. 318–327) (2006) 10. Wikipedia contributors, *Byzantine fault*. Wikipedia (2021). Retrieved August 8, 2021, from https://en.wikipedia.org/wiki/Byzantine_fault
11. J.R. Douceur, The Sybil attack, in *1st International Workshop on Peer-to-Peer Systems (IPTPS 02)* (2002)
12. *The Zilliqa Team. The Zilliqa technical whitepaper*. (n.d.). Zilliqa.Com. https://docs.zilliqa.com/whitepaper.pdf
13. *Cross Shard Transaction QuarkChain/pyquarkchain Wiki*. (n.d.). GitHub. Retrieved August 8, 2021. From https://github.com/QuarkChain/pyquarkchain/wiki/Cross-Shard-Transaction
14. E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, B. Ford, OmniLedger: a secure, scale-out, decentralized ledger via sharding. IEEE Symp. Secur. Privacy (SP), 583–598 (2018)
15. M. Zamani,M.Movhedi, R. Raykova, Rapidchain: scaling blockchain via full sharding. Conf. Comput. Commun. Secur., 931–948 (2018)
16. D. Sivaganesan, Performance estimation of sustainable smart farming with blockchain technology. IRO J. Sustain. Wireless Syst. **3**(2), 97–106 (2021)
17. E. Syta, P. Jovanovic, E. Kokoris-Kogias, N. Gailly, L. Gasser, I. Khoffi, M.J. Fischer, B. Ford, Scalable bias-resistant distributed randomness, in *38th IEEE Symposium on Security and Privacy* (2017)
18. G. Shvachych, I. Pobochy, E. Kholod, E. Ivaschenko, V. Busygin, Multiprocessor computing systems in the problem of global optimization, in *Structural Transformations and Problems of Information Economy Formation: Monograph* (Ascona Publishing, New York, USA, pp. 281–291) (2018)
19. G. Shvachych, V. Busygin, K. Tetyana, B. Moroz, F. Evhen, K. Olena, Designing features of parallel computational algorithms for solving of applied problems on parallel computing systems of cluster type. Inventive Comput. Technol. **191–200**(2019). https://doi.org/10.1007/978-3-030-33846-6_21
20. G. Shvachych, B. Moroz, I. Pobocii, D. Kozenkov, V. Busygin, Automated control parameters systems of technological process based on multiprocessor computing systems. Adv. Intell. Syst. Comput. **666–688**(2019). https://doi.org/10.1007/978-3-030-17798-0_53
21. G. Shvachych, N. Vozna, O. Ivashchenko, O. Bilyi, D. Moroz, Efficient algorithms for parallelizing tridiagonal systems of equations. Syst. Technol. **5**(136), 110–119 (2021). https://doi.org/10.34185/1562-9945-5-136-2021-11
22. S. Smys,H.Wang, Security enhancement in smart vehicle using blockchain-based architectural framework. J. Artif. Intelli. **3**(02), 90–100 (2021)