A Novel Consensus Protocol in Blockchain Network based on Proof of Activity Protocol and Game Theory

Zahra Boreiri*
Department of Engineering Science, College of Engineering
University of Tehran
Tehran, Iran
zahra.boreiri@ut.ac.ir

Alireza Norouzi Azad
Department of Engineering Science, College of Engineering
University of Tehran
Tehran, Iran
alireza.norouzi@ut.ac.ir

2022 8th International Conference on Web Research (ICWR) | 978-1-6654-6626-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICWR54782.2022.9786224

***Abstract*— Blockchain networks are already extensively used in various applications because of their increased security. The unique characteristics of blockchain technology, such as decentralized, peer-to-peer, and invariable distributed ledger qualities, make it appealing to researchers, academics, and industry. The consensus protocol is a fundamental part of blockchain technology. PoW (Proof of Work) or fixed-validator consensus protocols comprise most of the existing consensus mechanisms. However, the tremendous computational effort required for PoW leads to excessive energy and computing resource usage. On the other hand, Fixed-validator protocols validate new blocks by a fixed, static set of validators, allowing attackers to execute multiple attacks against these validators. In this article, we proposed a novel consensus protocol base on the Proof of Activity protocol and game theory. Our consensus protocol is efficient in energy consumption and can deal with selfish mining and majority-attack.**

Keywords—Blockchain; Game theory; Consensus protocols; Proof of Activity protocol; Selfish mining; Majority-attack;

I. INTRODUCTION

Satoshi Nakamoto first proposed the Blockchain concept in 2008, an invariable timestamp ledger of blocks [1]. In a distributed way, these blocks include records of transactions. Personal data, information regarding consensus protocols, payment history, and other information are all included in these transaction records. Blockchain technology in recent years has received a great deal of attention in various industrial sectors. The increasing number of cryptocurrencies being embraced daily demonstrates their significance. As of right now, over 2200 cryptocurrencies exist [1,2].

By eliminating centralized authority, every participant involved in the blockchain can share a distributed digital ledger of transactions. Users can contribute new blocks to the blockchain by completing computationally complex but readily verifiable challenges. The consensus method employed in the Blockchain network is based on this challenge, which prevents users from engaging in malicious activity. Since blockchain technology enables security, transparency, and decentralization in global peer-to-peer transactions, Blockchain technology has emerged as a viable option for a wide range of sectors in the past few years [3]. Regardless of whether it has the potential to improve security, new technologies always come with security concerns [4]. Blockchain technology has also enabled the development of block withholding attacks and selfish mining assaults. Many reasons drive these types of assaults, but the most common is financial gain. The game theory can be a valuable weapon in the fight against such assaults and in strengthening blockchain security.

A blockchain is a distributed, public ledger that consists of chained blocks, including several transactions. To ensure security, these blocks are verified worldwide and publicly. This validation must be conducted independently of a central authority. In order to verify, share, and synchronize the blocks among nodes, a decentralized, peer-to-peer, and distributed consensus process is used [5]. The consensus protocol is a fundamental element of blockchain technology. Before a block is added to the public ledger, a consensus mechanism on the blockchain must ensure that the whole network nodes agree on the validity of that block.

Additionally, the consensus protocol ensures that the order of blocks in the chain of each node in the blockchain network is the same. This guarantee is significant because blockchains are a network of decentralized nodes that need a way to synchronize their copies of information. The validators or miners are the nodes that are performing consensus protocols. There are several established consensus protocols. In a decentralized consensus protocol, an unknown, changeable number of validators must validate blocks to increase the stability of the protocol. However, all of the existing consensus protocols do not provide decentralization. Rather than that, they depend on predefined, well-known validators. This creates an opening for a variety of potential dangers [6].

Existing implementations of blockchain technology make use of a variety of consensus techniques, including Proof of Elapsed Time (PET), Proof of Stake Model (PSM), Stellar Consensus Protocol (SCP), Cross Fault Tolerance (XFT), Practical Byzantine Fault Tolerance Algorithm (PBFTA), Federated Byzantine Agreement (FBA), Ripple Consensus Protocol (RCP), Byzantine Fault Tolerance Algorithm (BFTA), etc.

The rest of this paper is organized as follows. In section II, we review some of the work related to the application of game theory in blockchain networks. Section III discusses how the Proof of Activity protocol works. Section IV presents the basic concepts of game theory. In section V, the proposed confusing protocol is discussed. Finally, the paper ends with summarizing and conclusion in section VI.

II. RELATED WORKS

Extensive research [7,8] has developed a consensus method in the Blockchain network. The authors in [9] compared blockchain consensus protocols. This study is conducted on

978-1-6654-6626-4/22/$31.00 ©2022 IEEE

various characteristics, including scalability, security, and rewarding validators issues. It has been shown that Proof of Work (PoW) is a widely recognized consensus method today. While blockchain technology has garnered great interest from the fields of computer science and economics, its application of game theory methodologies remains restricted [10]. In consensus procedures, game theory is also generally recognized. Numerous game theory-based consensus techniques are investigated, including the following:

The authors in [11] investigate what happens when pools begin to attack each other. Infiltrated miners from other pools may enter open pools and undertake withholding attacks. Open pools indicate pools of miners, which let any miner participate in mining operations. The mentioned article describes a game in which mining pools recruit members to enter other pools to reduce their mining power. There are two pools of players in the miner's dilemma game, and their strategies revolve around whether they attack each other or not. As this article points out, the most common tactic used by each player is attacking. Stone [12] provides a game-theoretic model with a block-size-dependent reward strategy. A big block size has a high payout cost. A threshold block size is explored in this section for a real-world situation. A greater block size introduces more latency, which isolates the block. Isolation is a significant contributing factor to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Additionally, it is noticed that the blockchain network's physical nature precludes a high gas price for a big block size.

Swanson [13] contends that in order to be secure, a permissionless collection of nodes in a blockchain network must use a game-theoretic method. The balance between blockchain network protocols and trust is a considerable difficulty, and numerous initiatives are being undertaken to progress certain areas on both sides. Block reward halving, in which miners' incentives are cut in half every four years, is not beneficial to participants since they want to rely on a network for longer than a decade. This might be accomplished by drastically raising transaction costs or lowering miners' security requirements (or both). Another option is to create a cost-effective and sophisticated game-theory model with a low entry price, robust security, and lengthy participant reliance. Dimitri [14] offers a block size-based mining game theory. A high block size raises the cost of computing resources used by a miner to generate a hash value indirectly. Additionally, the author establishes that the bitcoin mining process is lucrative when fewer independent individuals participate in mining, and that miners' rationality benefits everyone since big payoffs encourage miners to acquire extra computer capacity.

Liu et al. [15] conducted a thorough review of game theory applications on the blockchain. Non-cooperative games, extensive-form games, Stackelberg games, and stochastic games are some of the game theory models accessible, according to [15]. Selfish mining assaults, majority attacks, Denial of Service attacks, fraudulent data sharing, untrustworthy goods selling, and cyber-insurance may all be detected using game theory. Anyone in a blockchain network may employ token supply and transaction tipping while performing the role of blockchain mining, according to the Nakamoto protocol [16]. With the appropriate use of game theory, this method instills the profit process and optimizes the return. Game theory may also be used with two different types of mining management: individual mining and pool mining. Each miner's dominating tactic in the individual mining process is either processing power or forking the longest chain. Other activities in the pool mining-based game theory model for payout include block size setting, pool selection, and reward allocation.

According to Dey [17], there is a way that uses software agents to keep track of blockchain network participants. Algorithm game theory and supervised machine learning algorithms can be used to discover abnormalities as a result of this monitoring. In order to recognize different types of assaults, such as majority population size attacks, DoS, DDoS, etc., supervised machine learning is useful. Many alternative game theory-based models are used [18,19]. Among the evolutionary game models studied are hierarchical games and auctions. However, little research has been conducted on a single-game model for both resourceful and resource-constrained device networks. The primary difficulty in developing this model is the resource-dependent and challenge-variable technique of rewarding point-based game theory.

Indra Navaroj et al. [24], suggested an adaptive practical Byzantine fault tolerance consensus algorithm in permission blockchain network which splits each node into trust nodes and defective nodes. Voting is restricted to nodes with good reputations. In addition, the consensus process regards the node that was used to identify the trust as having a high reputation. The master node is chosen by a majority of voting values. This adaptive PBFT algorithm performs well in terms of long-term periodicity, greater scalability, and decreased total communication costs. Kovalchuk et al. [25], suggested a non-standard procedure for approving a transaction by a vendor to strengthen its security which it must wait for a specified number of confirmation blocks to be built, so to speak, under its supervision. Methods for calculating the upper estimates of this likelihood based on network factors and the amount of confirmation blocks that the vendor must view before deeming the transaction irreversible have been devised. The related numerical results are also provided for enhanced clarity. A significant characteristic of the obtained findings is that, in addition to the probability estimate, we can compute the number of confirmation blocks necessary to ensure the irreversibility of the transaction with a probability as close to 1 as desired.

The KRaft algorithm was proposed by Wang et al. [26] in 2019 which developed based on the Raft algorithm. By constructing a new node connection in the consensus protocol, the KRaft algorithm improves the election and consensus process. Scalability and transaction throughput improved by using many candidate nodes to replicate logs simultaneously. Based on the Raft algorithm, Wang et al. developed the hhRaft algorithm in 2021 [27]. This approach introduced a new monitor role to watch candidate and leader nodes, optimized the leader node election and log replication phases, increased fault tolerance, decreased consensus latency, and improved transaction throughput. In terms of anti-Byzantine failure capabilities, the hhRaft algorithm outperforms the Raft method, and it is better suited for high real-time and highly hostile situations.

DPoS, a highly centralized consensus method, was modified by Nir Bitansky [28] in 2020 to create more uncertainty in the voting process by including virtual nodes into the VRF consensus algorithm. The efficiency and low power consumption of DPoS are retained, but the algorithm's decentralization is also increased. Reijsbergen presented the LaKAS algorithm in 2021, which is a new form of PoS algorithm that significantly increases transaction speed and minimizes the danger of long-range assaults [29]. Changes to the accounting rights competition mechanism in the PoW algorithm suggested by Arjomandi-Nezhad [30] were made in 2021, and this modification was based on node donation contributions. The right to accounting becomes more readily available the more nodes give to the network as a whole. Social taxation situations, such as educational funds or charitable organizations, are more suited for this algorithm's use. Kara [31] proposed the CW-POW algorithm in 2021, which turned the PoW algorithm's single-round workload proof into a multi-round problem solution game, removing nodes that failed to solve the solution round by round, improving the algorithm's resistance to attacks and lowering its energy consumption significantly.

III. Proof of Activity

Direct attacks against Bitcoin's pure PoW system may be carried out in a variety of ways. The infamous >50% hashpower attack, in which the attacker invests in hardware equipment (ASIC) to gain more PoW hashpower than all other Bitcoin miners combined [8,20,21], is one kind of those attack. Such an attacker may then double-spend by reversing the recent ledger history to defraud merchants, or launch a PoW-denial-of-service (PoW-DoS) assault by refusing to include transactions in the blocks she creates, unless the transactions comply with the rules of attacker. This would allow such an attacker to either double-spend or carry out PoW-DoS attacks. Because double-spending or PoW-DoS attacks may undermine trust in the Bitcoin system, the attacker may succeed in her goal if she intends to do damage. Using PoW-DoS to extort others and raise transaction fees is an option for a selfish self-interested attacker who wants to profit from double-spending.

PoW-DoS attacks would increase if the attacker has at least 50% of the total hashpower. Additionally, an attacker who acquires a significant proportion of the overall hashpower may undertake double-spending attacks [8,21]. The nodes in the network of Proof of Activity protocol (PoA) do more complicated verifications in comparison to the nodes in the network of Proof of Work protocol (PoW). Follow-the-satoshi, the principal PoA subroutine, picks a single satoshi (the smallest unit of bitcoin) at random from a pool of all the Satoshis that have ever been created. Once the pseudorandom index is selected, it is followed by analyzing the block where this Satoshi was created, and then following each subsequent transaction that moved the coin to the next address until it reaches the current owner of the Satoshi. We specify how blocks are generated in the PoA network as follows:

Firstly, each miner puts their computing resources into creating an empty block header, which is made up of just the hash of the previous block and a nonce as well as their own public IP address and height concerning the genesis block. This header has no connection to any transactions.

Secondly, when a miner has generated an empty block header indicating that the hash of a miner's data is lower than the current difficulty goal, the miner broadcasts the block header in the network.

In the third step, the hash of this block header is used as data by all network nodes, which generates N pseudorandom stakeholders deterministically. Each combination of the preceding block's hash and N fixed suffix values are concatenated with this one and hashed before being used as input to the follow-the-Satoshi function.

In the fourth step, every online stakeholder examines whether the miner is broadcasted empty block header is legitimate, which means it includes the hash of the previous block and satisfies the current difficulty. Following validation, the stakeholder determines if she is one of the N fortunate stakeholders for this block. When the first N-1 fortunate stakeholders realize that the block is derived from them, they sign the hash of this empty block header with the private key that controls their derived Satoshi and broadcast their signature to the network. When the N^{th} stakeholder notices that the block derives her, she constructs a wrapped block that expands the empty block header by containing as many transactions as she wants, the N-1 signatures of the other derived stakeholders, and her own signature for the hash of the whole block.

In the fifth step, The N^{th} stakeholder broadcasts the wrapped block to the network, and when the other nodes discover that it is genuine according to the above, they consider it a legitimate extension of the blockchain. The nodes attempt to extend the most extended branch of the blockchain they are aware of, where "longest" is measured in PoW difficulty, similar to Bitcoin.

Finally, the miner and the N fortunate stakeholders split the fees from the transactions that the Nth stakeholder earned [22].

The PoA protocol tries to decentralize the authority that synchronizes transactions in a significant way. To dominate the block production process, an attacker must hold a significant portion of the total quantity of bitcoin created so far. In most cases, the cost of an attack using the PoA protocol would be substantially greater than with Bitcoin's pure PoW algorithm. Furthermore, the PoA protocol is anticipated to achieve other advantageous qualities, such as enhanced network architecture, incentives for keeping fully operational nodes, minimal transaction costs, and more efficient energy utilization [22].

IV. Game Theory

A game is a tuple with a finite set of players $\mathcal{N} = \{1, \dots, N\}$; a finite set $\mathcal{A}_k$ of strategies for each player $k \in \mathcal{N}$. The set of strategy profiles of the game is $\mathcal{A} \overset{\text{def}}{=} \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \mathcal{A}_N$. The payoff functions of the players are defined as $u_k: \mathcal{A} \to \mathbb{R}$. The best response correspondence $\mathrm{br}_k(x)$ as the set of all strategies that maximizes the payoff for player k under profile $x = (x_1, \dots, x_N)$ is shown at (1).

$$\mathrm{br}_k(x) \overset{\text{def}}{=} \left\{ \underset{\alpha \in \mathcal{A}_k}{\mathrm{argmax}}\, u_k(\alpha; x_{-k}) \right\}. \tag{1}$$

A Nash equilibrium (NE) is a profile x^* such that $x_k^* \in \mathrm{br}_k(x^*)$ for every player k [23].

V. The Proposed Consensus Protocol

In the POA algorithm, the node which wants to add a block must solve the same problem as in the POW, which requires massive computational effort, high cost, and special hardware. In this paper, we modify the POA algorithm so that nodes do not tend to violate the protocol without having to solve a difficult problem based on a model in game theory.

Like the Proof of Stake (PoS) protocol, each node can be selected as the leader based on the amount of cryptocurrency it has, and its validity rate but this is done randomly, so no node knows when this will be possible. In this algorithm, the nodes that want to be elected as leaders block some of their cryptocurrencies. In fact, from the nodes that have the most validity rate, the one that has blocked the most cryptocurrencies will be selected as the leader in the next step. In order not to know in advance which node is the winner, the choice of leader is taken out of the final state and is done to some extent randomly. It can be very effective in reducing the likelihood of fraud. Here we use a combination of hash code and stock. Therefore, the lower the node hash code and the larger the stock, the higher the chance of selection. This algorithm is performed in the following steps:

1. If a node wants to be elected leader in the next step, based on its validity rate it blocks part of its cryptocurrencies and notifies the network.

2. From among the candidate nodes, the node that has blocked most cryptocurrencies is selected as the leader.

3. The leader randomly selects N nodes, where N is a random number. In this way, N random numbers are generated between one and the maximum number of Satoshis in the network. Since a Satoshi may have been traded many times, a node is considered to be the owner of the Satoshi who is the last owner. Then the leader node sends the set of selected Satoshis to all network nodes. We call this set of nodes, acceptors.

4. Whenever a node in the network receives this message, it checks whether it owns one of the selected Satoshis. Satoshi owner's node then participates in the process as one of the nodes selected by the leader.

5. The leader node creates a new non-transactional block that contains the hash code of the previous block, the general address of the leader, and the distance from the first block of the chain, then sends it to all nodes in the network. Whenever an acceptor receives this message, it first checks that the block has been created correctly and has complied with the rules. In fact, they examine the header of the empty block formed by the leader. If it was valid and had registered the hash code of the previous block correctly, then they sign the blank block hash code and notifies the leader. Since the leader places the block header and it blocks the most stocks and has the most validity rate, it is very unlikely that the node is an attacker and mistakenly adds the previous block hash code and other information to the new empty block.

6. Acceptor nodes may be members of mining pools, so they play a strategic game modeled as follows.

Players = {Mining pool member nodes, Individual nodes}

Strategies for each player = {Approve, Disapprove}

In the following, we show each strategy profiles as (A, B) in which A represents the action of the mining pool member nodes and B represents the action of the individual nodes. The strategy profile (Disapprove, Disapprove) means the added empty block is really wrong and should not be verified, and also the attacker nodes that form the mining pool do not approve it. The strategy profile (Approve, Approve) means the added empty block is really valid and should be verified, and also the attacker nodes that form the mining pool approve it. The strategy profile (Disapprove, Approve) means the added empty block is really valid and should be verified, but the attacker nodes that form the mining pool do not approve it. The strategy profile (Approve, Disapprove) means the added empty block is really wrong and should not be verified, but the attacker nodes that form the mining pool approve it. Now we determine payoff functions or utility function for each players.

As it can be seen in (2) and (3) the u_M and u_I demonstrate the utility function for Mining pool member nodes and Individual nodes, respectively. Let A and D illustrate approval and disapprove, respectively.

$$u_M(\mathrm{D,D}) > u_M(\mathrm{A,D}) > u_M(\mathrm{A,A}) > u_M(\mathrm{A,D}) \quad (2)$$

$$u_I(\mathrm{D,D}) = u_I(\mathrm{A,A}) > u_I(\mathrm{D,A}) > u_I(\mathrm{A,D}) \quad (3)$$

As it can seen, the worst-case scenario for any group of nodes is when that group disapproves the new block while the rival group approves the block. The Nash equilibriums are (Disapprove, Disapprove) and (Approve, Approve). This happens when both groups have an honest strategy; that is, if the new block is added wrongly, they disapprove the block, and if it is completely correct, they approve it.

7. When a disagreement occurs between nodes, the leader node puts the nodes that disapproved the block in a list called the blocked nodes list and adds the block. The leader then sends the set of blocked nodes to all the nodes in the network. In this case, the validity rate of these nodes decreases, which means that other nodes no longer trust these blocked nodes to be selected as the leader.

8. Now one of the acceptor nodes which is not in the blocked nodes list, can be taken as the block mining node. According to (4), a node is the block mining node whose hash function value is on the sum of the previous block's hash code and the general node address of the maximum value.

$$max\{hash(hash(\text{previous block}) + \text{the general node address})\} \quad (4)$$

9. In the next step, block mining node, adds the desired transactions to the block and practically forms a new block then sends that throughout the network.

10. In the last step, the reward regarding block mining is divided between the leader node and the block mining node in the last step. Therefore, the nodes do not tend to form mining pools and attack the network because they lose the chance to be an addition and a leader and can not get a reward.

VI. Conclution

In this paper, we introduce a novel consensus protocol in blockchain networks which Its Energy consumption is very low, Block creation speed is very high and not Requires advanced hardware. In this protocol no nodes tend to forming mining pools. Any node that has blocked most cryptocurrencies and not being in the block node list is selected as the leader. The leader selects *N* random numbers between one and the maximum number of Satoshis in the network. The leader sends a set of Satoshis to network. Whenever a node in the network receives the set of Satoshis, it checks whether it owns one of the selected Satoshis. We call the owner of the Satoshis acceptors. The leader creates a new non-transactional block and sends it to all nodes in the network. Whenever an acceptor receives the empty block, it first checks that the block has been created correctly, then it signs the blank block hash code and notifies the leader. Since the leader places the block header and blocks the most stocks, and has the most validity rate, it is improbable that the node is an attacker and mistakenly adds the previous block hash code and other information to the new empty block. Acceptor nodes may be members of mining pools, so they play a strategic game. According to Nash equilibrium, we showed that the whole nodes in the networks do not tend to create mining pools. Then one of the acceptor nodes, which is not in the blocked nodes list, which we call the block mining node, adds the transactions to the block and creates a new block, then sends that throughout the network. Finally, the block mining reward is divided between the leader and the block mining node. Therefore, the nodes do not tend to form mining pools and attack the network because they lose the chance to be an addition and a leader and can not get a reward.

In general, consensus algorithms are assessed based on three criteria: performance, security, and the degree of decentralization. The most critical considerations in determining system performance are scalability, complexity, throughput, and algorithm activity. For the most part, anti-Byzantine ability and resistance to double-spending are responsible for security. Also, the scale of nodes involved in the consensus process determines the decentralization level. The protocol proposed in [32] improved the scalability since it uses a pipelined block topology structure and there is no necessary sequence link between the production of blocks. Because the algorithms proposed in [33] and [34] are based on standard BFT, there is no scalability increase. Point-to-point propagation is used in all algorithms in [33], [34], and [35], hence the communication complexity is O(n2). The communication complexity in the literature [32] is O(n) since it utilizes a star topology. Concurrency is low in [32], [33], and [34], which all employ the classic chain structure. In comparison, the proposed protocol in [32] uses a pipeline approach and does not impose a time constraint on the production of the front and back blocks, allowing for more concurrency. The authors of [32], [33], [35], and [34] utilized a chained bottom topology, with the lowest unit of operation being a block, ensuring that their algorithms are resistant to double spending. When it comes to fault tolerance, the proposed consensus algorithms in [32], [33], and [34] are based on the maximum fault tolerance of BFT. In comparison to the prior literature, our algorithm is high concurrency also the scalability of suggested algorithm and resistance to double spending are quite strong, while the communication complexity is O(n).

In Table I., we compared the property of our proposed consensus protocol to other common protocols.

Acknowledgment

The authors like to express their gratitude to the Editors and anonymous reviewers for their helpful remarks and ideas.

TABLE I. Comparison of our proposed protocol with other protocols

	POW	***POS***	***POA***	***DPOS***	***PBFT***	***Proposed protocol***
Energy consumption	Very high	Low	Very high	Very low	Very low	Very low
Selfish Mining and Majority-Attack	Yes	Yes	Yes	Yes	Yes	No
Possibility of forming mining pools	Yes	Yes	Yes	Yes	Yes	No
Block creation speed	Very low	High	Very low	Very high	High	Very high
The degree of decentralization	Very high	Very high	High	High	Very low	Very low
Requires advanced hardware	Yes	No	Yes	No	No	No

REFERENCES

[1] D. Xu, L. Xiao, L. Sun and M. Lei, "Game theoretic study on blockchain based secure edge networks," 2017 IEEE/CIC International Conference on Communications in China (ICCC), pp. 1–5, 2017.

[2] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557–564, 2017.

[3] M. Pilkington, "Blockchain Technology: Principles and Applications," Research Handbook on Digital Transformations, 2016.

[4] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, T. Moore, "Game-theoretic analysis of DDoS attacks against bitcoin mining pools" Financial Cryptography and Data Security - FC 2014 Workshops, BITCOIN and WAHC 2014, Springer Verlag, Vol. 8438, pp. 72–86, 2014.

[5] T. K. Mackey, G. Nayyar, "A review of existing and emerging digital technologies to combat the global trade in fake medicines," Expert opinion on drug safety, 16(5), 587–602, 2017.

[6] N. Alzahrani, N. Bulusu, "Towards True Decentralization: A Blockchain Consensus Protocol Based on Game Theory and Randomness," International Conference on Decision and Game Theory for Security, GameSec 2018: Decision and Game Theory for Security, pp. 465–485, vol. 11199, 2018.

[7] W. Wang et al., "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," in IEEE Access, vol. 7, pp. 22328–22370, 2019.

[8] N. Satoshi, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2009.

[9] L. M. Bach, B. Mihaljevic and M. Zagar, "Comparative analysis of blockchain consensus algorithms," 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1545–1550, 2018.

[10] M. Nojoumian, A. Golchubian, L. Njilla, K. Kwiat, C. Kamhoua, "Incentivizing blockchain miners to avoid dishonest mining strategies by a reputation-based paradigm," IEEE Computing Conference (CC). IEEE, London (2018)

[11] I. Eyal, "The Miner's Dilemma," 2015 IEEE Symposium on Security and Privacy, pp. 89–103, 2015.

[12] A. Stone, "An examination of single transaction blocks and their effect on network throughput and block size," Self-published Paper, 2015.

[13] T. Swanson, "Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems, " Report, 2015.

[14] N. Dimitri, "Bitcoin Mining as a Contest", ledger, vol. 2, pp. 31–37, Sep. 2017.

[15] Z. Liu et al., "A survey on applications of game theory in blockchain." arXiv, 2019.

[16] A. Kumar, S. Jain, "Proof of Game (PoG): A Game Theory Based Consensus Model," In: Karrupusamy, Sustainable Communication Networks and Application(ICSCN), Lecture Notes on Data Engineering and Communications Technologies, vol. 39, 2020.

[17] S. Dey, "Securing Majority-Attack in Blockchain Using Machine Learning and Algorithmic Game Theory: A Proof of Work," 10th Computer Science and Electronic Engineering (CEEC), pp. 7–10, 2018.

[18] X. Liu, W. Wang, D. Niyato, N. Zhao and P. Wang, "Evolutionary Game for Mining Pool Selection in Blockchain Networks," in IEEE Wireless Communications Letters, vol. 7, no. 5, pp. 760–763, Oct. 2018.

[19] Y. Jiao, P. Wang, D. Niyato and Z. Xiong, "Social Welfare Maximization Auction in Edge Computing Resource Allocation for Mobile Blockchain," 2018 IEEE International Conference on Communications (ICC), 2018.

[20] S. Barber, X. Boyen, E. Shi, E. Uzun, "Bitter to Better — How to Make Bitcoin a Better Currency," In: Keromytis, A.D. (eds) Financial Cryptography and Data Security, Computer Science, vol. 7397, pp. 399–414, 2012.

[21] M. Rosenfeld, "Analysis of Hashrate-Based Double Spending," arXiv, 2014.

[22] I. Bentov, Ch. Lee, A. Mizrahi, M. Rosenfeld, "Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract]y," SIGMETRICS Perform, Eval. Rev. 42, 3, pp. 34–37, 2014.

[23] S. Durand, B. Gaujal, "Complexity and Optimality of the Best Response Algorithm in Random Potential Games," Symposium on Algorithmic Game Theory (SAGT), pp. 40–51, 2016.

[24] G. Indra Navaroj, E. Golden Julie, Y. Harold Robinson, "Adaptive practical Byzantine fault tolerance consensus algorithm in permission blockchain network," International Journal of Web and Grid Services, Vol. 18, No. 1, 2021.

[25] L. Kovalchuk, R. Oliynykov, Y. Bespalov, M. Rodinko, "Comparative Analysis of Consensus Algorithms Using a Directed Acyclic Graph Instead of a Blockchain, and the Construction of Security Estimates of Spectre Protocol Against Double Spend Attack," Information Security Technologies in the Decentralized Distributed Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 115, pp. 203-224, 2022.

[26] R. Wang, L. Zhang, Q. Xu and H. Zhou, "K-Bucket Based Raft-Like Consensus Algorithm for Permissioned Blockchain," 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), pp. 996-999, 2019.

[27] Y. Wang, S. Li, L. Xu, L. Xu, "Improved Raft Consensus Algorithm in High Real-Time and Highly Adversarial Environment," International Conference on Web Information Systems and Applications; Springer: Cham, Switzerland, pp. 718–726, 2021.

[28] B. Nir, "Verifiable Random Functions from Non-interactive Witness-Indistinguishable Proofs," J. Cryptol, vol. 33, 459–493, 2020.

[29] D. Reijsbergen et all., "A Probabilistic Proof-of-Stake Protocol," Proceedings 2021, Network and Distributed System Security Symposium; Internet Society: Reston, VA, USA, 2021.

[30] A. Arjomandi-Nezhad, M. Fotuhi-Firuzabad, A. Dorri, P. Dehghanian, "Proof of humanity: A tax-aware society-centric consensus algorithm for Blockchains," Peer—Peer Netw. Appl, vol. 14, pp. 3634–3646, 2021.

[31] M. Kara et all., "A Compute and Wait in PoW (CW-PoW) Consensus Algorithm for Preserving Energy Consumption," Appl. Sci , 11, 6750, 2021.

[32] I. Abraham, D. Malkhi, and K. Nayak, "Sync HotStuff: simple and practical synchronous state machine replication," in Proceedings of the 41th Symposium on Security and Privacy, Piscataway, May 2020.

[33] M. T. d. Oliveira, L. H. A. Reis, D. S. V. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. F. Mattos, "Blockchain reputation-based consensus: a scalable and resilient mechanism for distributed mistrusting applications," Computer Networks, vol. 179, no. 10, pp. 107367–107380, 2020.

[34] N. A. Lin, Z. H. Chen, and G. K. Liu, "Mechanism for proof-of-reputation consensus for blockchain validator nodes," Journal of Xidian UniverSity, vol. 47, no. 5, pp. 61–66, 2020.

[35] H. Liu, S. Li, and W. Lv, "Master-slave multiple-blockchain consensus based on credibility," Journal of Nanjing University of Science and Technology, vol. 44, no. 3, pp. 325–331, 2020.