Peer-to-Peer Networking and Applications (2021) 14:3634–3646
https://doi.org/10.1007/s12083-021-01204-4

Proof of humanity: A tax-aware society-centric consensus algorithm for Blockchains

Ali Arjomandi-Nezhad[1] · Mahmud Fotuhi-Firuzabad[1] · Ali Dorri[2] · Payman Dehghanian[3]

Received: 30 October 2020 / Accepted: 3 June 2021 / Published online: 30 June 2021

Abstract

Blockchain technology brings about an opportunity to maintain decentralization in several applications, such as cryptocurrency. With the agents of a decentralized system operating independently, it calls for a consensus protocol that helps all nodes to agree on the state of the ledger. Most of the existing blockchains rely on Proof of Work (PoW) as the underlying consensus algorithm, resulting in a significant amount of electricity power consumption. Furthermore, it demands the miner to buy specific computation devices. Besides, a protocol to gather the society-related taxes such as public education funding and charities is lacking in existing consensus algorithms. In response, this paper proposes a new consensus algorithm, namely Proof of Humanity (PoH) aiming at gathering society-related taxes. According to PoH, the probability that an agent becomes a leader depends on its donations to non-profit accounts. Therefore, PoH encourages miners to donate money and gain mining power, its incentives, and transaction fees. The associated bureaucracy model is introduced briefly to address the required ecosystem for real case implementation of PoH. A distributed random variable generation algorithm is presented in this paper which ensures that the randomly selected leader is neither predictable nor adjustable. It is demonstrated that the proposed blockchain is totally robust against forking and possesses a high level of propagation speed, which ensures the scalability. Simulations show that the proposed blockchain network does not fail even in adverse scenarios where the majority of nodes refuse to propagate valid blocks. Besides, simulations reveal a suitable average block creation duration.

Keywords Blockchain · Proof of Humanity · Power consumption reduction · Consensus · Society-related tax

✉ Mahmud Fotuhi-Firuzabad
fotuhi@sharif.edu

Ali Arjomandi-Nezhad
a.arjomandi@alum.sharif.edu

Ali Dorri
ali.dorri@qut.edu.au

Payman Dehghanian
payman@gwu.edu

[1] Sharif University of Technology, Azadi Ave., P.O.Box 11155-4363, Tehran, Iran

[2] Queensland University of Technology, 2 George St, Brisbane, QLD 4000, Australia

[3] George Washington University, 800 22nd St. NW, Washington, DC 20052, USA

1 Introduction

In recent years, blockchain has attracted tremendous attention since its introduction in Bitcoin [1], the first cryptocurrency, due to its salient features, including the elimination of authority of intermittency and trustful framework. Blockchain ensures that the participating nodes can trust untrusted participants and agree on the state of a ledger without relying on any central authority (e.g., bank) by employing a consensus algorithm. The latter is defined as "the process of agreement between distrusting nodes" [2]. The term *distrusting* is mentioned because there is no trust between anonymous nodes, especially in the case of digital currency which involves money ownership. Therefore, being decentralized and endowing with a consensus algorithm makes the blockchain highly secure. All of the participants in the blockchain network are recognized with a changeable address, which does not disclose their real identity. Blockchain is defined as "a distributed ledger technology" [3]. It provides an immutable trace of transactions and contracts that can be tracked by any network's participant. Certain features of blockchain technology including transparency, security, and scalability make it a sufficiently effective replacement to the existing financial frameworks. Existing bank-based financial bureaucracy system frameworks are envisioned to be supplanted with blockchain-based cryptocurrency in the near future [4]. From the early emergence of money until now, the financial bureaucracy, i.e. information processing machines [4], has had the following main evolutions (see Fig. 1):

1) Monarch-based: A king or an emperor issued money and was in charge of gathering taxes and preventing forge.
2) Bank-based: In the current modern financial bureaucracy, banks are responsible for issuing money, keeping track of transactions, taxes, etc.
3) Blockchain-based: The upcoming financial bureaucracy is Internet-based and encrypted. Any third party (monarch or banking system) will be bypassed.

Blockchain's participants' identities are anonymous, and thereby it is hard to track a user's expenses or income which is critical for governments to enforce the tax. Besides, several big companies are accepting cryptocurrencies such as Bitcoin, which makes it even more challenging for governments to calculate their taxes. Therefore, with the domination of blockchain-based bureaucracy, many big issues arise. Public health, public education, road construction, and city maintenance would face a lack of budget. Besides, there would be no tax to support the lower-income citizens and communities who need charity supports. One solution to address this challenge is to intrinsically incentivize the blockchain participants to donate money to organizations which spend it on not-for-profit goals such as helping poor communities, road construction, health services, funding public education, etc. The proposed model is called *society-centric blockchain-based* bureaucracy. A blockchain consensus algorithm, which is introduced in this paper, is required to realize this bureaucracy. In addition to the aforementioned economic problem, there are some technical issues associated with the contemporary consensus algorithms. High resource consumption, fork creation, which imposes the risk of double-spending a value, and the long delay in block creation are the principal challenges that should be addressed meanwhile designing society-centric blockchain consensus.

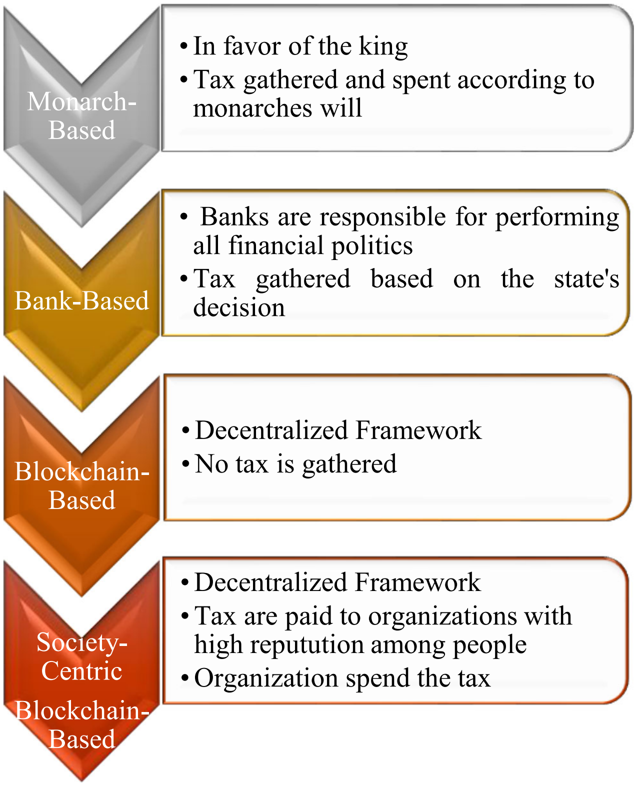

Monarch-Based
• In favor of the king
• Tax gathered and spent according to monarches will
Bank-Based
• Banks are responsible for performing all financial politics
• Tax gathered based on the state's decision
Blockchain-Based
• Decentralized Framework
• No tax is gathered
Society-Centric Blockchain-Based
• Decentralized Framework
• Tax are paid to organizations with high reputution among people
• Organization spend the tax

Fig. 1 The main features in the past, present, future, and the proposed financial bureaucracy

1.1 Contributions

The main contribution of this paper is to address the outlined challenge by designing a society-centric consensus algorithm for cryptocurrencies. We address the society-related tax taking problem by proposing a new consensus algorithm. To the best of our knowledge, no research has been done to involve the society-related taxes in consensus algorithms. This need is driven by the society's economics rather than technical essences. Motivated by the aforementioned challenges, this paper proposes the Proof of Humanity (PoH) as a leader selection consensus algorithm. The nodes that donate to the organization accounts are grouped as leader candidate nodes, among which the leader is chosen. The word *organization* refers to not-for-profit entities that conduct positive society-related activities such as public health and welfare. This framework provides the society, the power grid, and the environment with several merits:

1. Society, especially poor communities, would benefit from this structure. This is because some amount of money is spent on their service. As the competition between candidate leaders always runs up and new candidate leaders enter into the battle, there is always some payments from candidate leaders into the donation pool.
2. By implementing PoH, a huge amount of energy, which was previously consumed by mining pools, will be cut off. As a result, the pressure of this demand on the electric utility will reduce. Furthermore, the environmental concerns arisen by burning fossil fuels in power plants decrease.
3. Since the need for solving a puzzle has been eliminated, computational resources, e.g. ASICs, are no longer

required for block mining. Therefore, the demand for processors will decline. In consequence, processors will be available at lower prices for other purposes.

4. In PoW based blockchains, there was a long delay for each block creation, which reduce the scalability of the whole blockchain. PoH eliminates this unnecessary delay.

The probability that a candidate leader becomes the leader is proportional to the money that the candidate leader donated to the organization accounts and the reputation of the organization account that received the donation. In the calculation of the Donations Share (DS), all donations are weighted with the community's trust in the organization. In other words, miners would invest parts of the money that they earned from taking transaction fees to the organization pool rather than in ASIC devices which consume a huge amount of energy. In this way, not only huge resource consumption is cut, but also the long block creation delay, which is caused previously by solving PoW puzzle, is bypassed. By enforcing the next leader's address right after the creation of each block, the fork creation would be impossible. The leader selection algorithm is a random process with the probabilities mention earlier in this paragraph.

A main part of consensus algorithm design is the decentralized random number generation. We propose a new random number generator algorithm. If the generated random number is neither predictable nor adjustable, the blockchain would be robust to leader selection manipulation. The randomness of the generated variable will be discussed in detail. If the chosen leader does not send the block for more than 20 min (known as the creation deadline), it will be assumed that the leader is inactive and another random variable is chosen. Then, the propagation prohibition, which is the only resilience threat for the proposed blockchain, is investigated. In this threat, malicious and lazy nodes prohibit the propagation of blocks which (a) decreases the scalability of the blockchain, and (b) makes the leader unsuccessful in creating the block within a creation deadline. We study the resilience of the proposed blockchain by generating thousands of instances of simulations and computing the number of nodes that received a valid block in each of them. Simulations prove that the proposed consensus algorithm is highly resilient even in cases where a vast majority of nodes are malicious or lazy and do not pass the blocks to neighbor nodes. The resilience in this context reflects that all nodes will receive the block created by a leader within the creation deadline duration. Block creation duration, which is the average time between inclusions of two successive blocks, is also simulated and quantified. Results will indicate a fast block creation process which, in consequence, enhances the scalability of the blockchain.

To summarize, Fig. 2 depicts the five main stages toward a society-centric blockchain with PoH consensus algorithm. These stages are: (1) realizing the essence of the proposed framework, (2) designing the required bureaucracy of this financial system, (3) designing the holistic consensus algorithm, (4) designing the technical details and algorithms, and (5) programming and practical test. While the first stage has been explored in details in Sections 1 and 2, the second and third stages would be discussed in subsections 4.1 and 4.2. The designed consensus algorithm not only incentivizes tax donations but also prevents fork creation and leader selection manipulation. The last two stages of Fig. 2 entail several technical details such as encryption algorithm, maximum block size (or gas), virtual machine structure, hashing algorithm, which need to be addressed in future research.

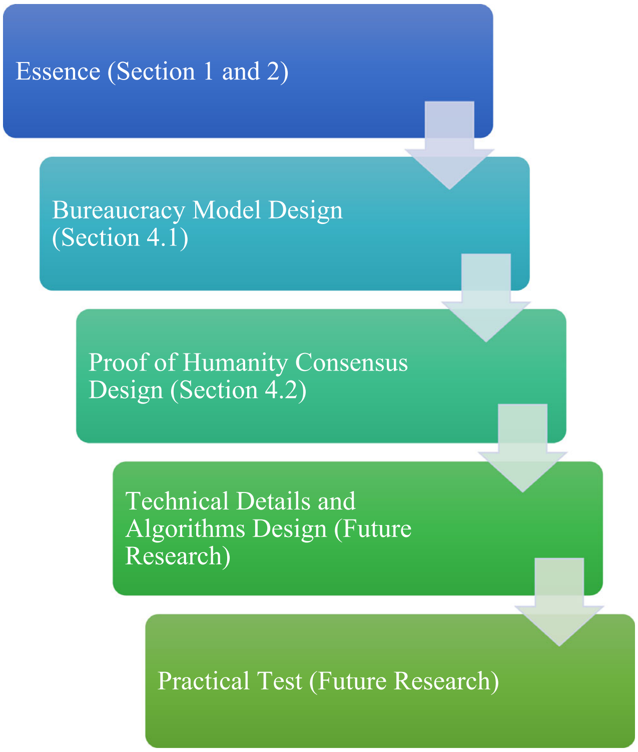

Essence (Section 1 and 2)
Bureaucracy Model Design (Section 4.1)
Proof of Humanity Consensus Design (Section 4.2)
Technical Details and Algorithms Design (Future Research)
Practical Test (Future Research)

Fig. 2 The pathway to implement Society-Centric Blockchain

The rest of this paper is organized as follows. The state-of-the-art literature on the consensus algorithms is reviewed in Section 2. Section 3 briefly reviews the most important concepts involving blockchain. The general bureaucracy model and the proposed PoH consensus methodology are introduced in Section 4. The resilience concerns and measures to resilience enhancement are discussed in Section 5. The duration of block's creation, computational overhead, and the resilience of the proposed blockchain are evaluated in Section 6. Finally, conclusions are drawn in Section 7.

2 Related works

There are many decentralized applications for blockchain technology, such as smart city [5], IoT [6, 7], and data storage [8]. Since this paper mainly focuses on the financial application of the blockchain, which is cryptocurrency, we here review only the consensus algorithms related to this specific niche of research. Besides, developing the cryptographic aspect of blockchain [9–12], i.e. building encryption algorithms, is beyond the scope of this paper.

One of the earliest consensus algorithms is the practical byzantine fault tolerance (PBFT) consensus algorithm [13]. The algorithm works properly if the number of faulty nodes is less than one-third of all nodes. An optimized PBFT is presented in [14] in order to enhance the scalability of the PBFT-based blockchains. Nevertheless, this paper does not solve the main pitfall of PBFT. Among various BFT algorithms, Helix is developed to ensure fair order of transactions [15]. In this schema, transactions are encrypted using threshold encryption to hide the transactions' information from nodes which may benefit from manipulating transaction ordering. Proof of Work (PoW) is the consensus algorithm of the most well-known practical blockchain, which is Bitcoin [1]. This algorithm is used in the current version of the Ethereum digital currency as well [16]. The fundamental rule of this algorithm is that if multiple forks exist, the one with the biggest chain is valid. The proof of works requires miners to solve a hard-to-solve yet easy to verify puzzle which in turn consumes significant resources. The principal idea behind this algorithm is that a fraud miner who wants to double-spend a transaction (in Bitcoin) or to double-spend a value (in Ethereum) should own 51% of the network computational power in order to create a valid fork. Otherwise, it would be practically impossible for him/her to make a valid fork. Due to the great number of miners and their numerous ASIC mining devices, no single miner can attack the system by building a cryptographically valid non-trust fork. Despite the successful performance of PoW in decreasing the attack risk, it typically results in a huge amount of electrical power consumption which, in turn, harms the environment and also imposes significant costs. In response, several other consensus algorithms have been proposed to overcome this challenge. Proof of Authority (PoA) is a preliminary solution for eliminating the need for a high amount of energy for block creation. PoA relies on a set of trusted nodes. At least half of the authorized nodes should be honest [17]. Another consensus algorithm that is suggested in the literature is the Proof of Burn (PoB) [18]. This algorithm suggests that a candidate leader who destroys more money through a special kind of transaction is more likely to be chosen as the leader. PoB implies that miners spend their digital assets directly instead of paying bills for consumed electricity. Although this approach significantly cuts off electricity consumption, the burned money does not have any merits for anyone in society.

One of the most popular consensus algorithms is the Proof of Stake (PoS) [19]. According to PoS, there will be no miner. Instead, the blocks are validated via a validation process. Each digital coin possesses a serial number. A random number is generated via a seed algorithm. The account which possesses the coin that has the same serial number as the randomly-generated number is considered as a validator [20]. Thus, accounts with more money would be more probable to be a validator (elector). If a validator's opinion about the correction of a block is in accordance with the other validators, he/she would be given some amount of block reward; otherwise, he/she loses all of his/her digital property. Therefore, for the sake of his/her share of reward, he/she will act honestly. There are several versions of the PoS consensus algorithms, such as the Ouroboros, Chain of Activity, Casper, Algorand, and Tendermint [20]. While they all follow the same concept of leader/validator selection according to the stake, they are differentiated with respect to the random number seed algorithm and the interaction between the leader and the validators. This paper briefly reviews the Chain of Activity (CoA) because some of its techniques are used in this paper as well. According to CoA, headers of the current l blocks seed a random number for choosing leaders of the following l blocks [21]. As a result, fork creation, which results in a double-spending attack and is a threat for PoW-based blockchains, is impossible. The general idea of preventing fork by seeding from previous blocks is adopted in our paper. However, there might be collusion challenges regarding the random number seeding of this algorithm. The leaders of the current l blocks may decide to choose their blocks' transactions in a way that leaders of the upcoming l blocks be chosen desirably. Including transactions in such a way that a set of desired accounts become leaders is computationally hard. It, however, is theoretically possible; especially with the ever-growing computational advancements. Therefore, a new random number seeding method is used in our paper. The main problem regarding PoS algorithms is that the rich gets richer.

A voting-based consensus algorithm for consortium blockchain is presented in [22]. Nodes play different roles, i.e. verifier, packager, candidate, in the block creation process. Any activity in the block creation process is rewarded. In contrast, any behavior violating the integrity of the blockchain would be punished either in reputation or asset. Again, this consensus algorithm does not suggest any solution for tax gathering in the blockchain framework.

Recently, reference [23] presented a mixed clustering and PoS consensus algorithm for charity donation tracking blockchain. This reference, despite its merit, lacks some key points that motivated our research:

- Gathering charity donations is not sufficient in the era of blockchain-based bureaucracy. Some money for both charity and public spends such as public education, should be gathered.
- Developing a separate blockchain for donation purposes will not result in a sustainable flow of donations. Instead, the blockchain of the main cryptocurrency used in

individual's daily trades should change in a way which encourages people to donate (to either charities or public spending)

3 Blockchain background

In this section, we provide a background discussion on blockchain. Blockchain is a database shared across all participating nodes where data is grouped in the form of blocks. The participating nodes exchange information and communicate by generating transactions [19]. Each transaction contains a Public Key (PK) that is used as the identity of the transaction generator which in turn introduces high anonymity for the blockchain participants. The transaction generator signs the hash of the transaction content using the corresponding private key. Hash function converts an input with any size to a fixed-size output. With a change in one bit of the input, the output completely changes. Technically, it is impossible to find two input values leading to the same output or finding the input with knowledge on the output of a hash function. Therefore, signing the hash of the transaction content ensures the transaction's integrity and non-repudiation. Hash of the public key of each account is the address of that account. In each transaction, the address of the recipient is mentioned.

New transactions are broadcasted and verified by the participating nodes. Particular nodes in the network, known as leaders (also as miners and validators in the literature) collect new transactions, form a new block, and append it to the blockchain following a consensus algorithm. The latter introduces randomness among the leaders and limits the number of blocks that can be appended to the blockchain and, thus, is the key to blockchain security. The leader is paid a specific amount of block incentive plus transaction fees taken from transaction generators.

It is depicted in Fig. 3 that each block contains a header which is the hash of all the contents of the block, the hash of previous data, the public key of the block creator, metadata, and transactions. The metadata consists of the reward, block number, timestamp, size, etc. The hash of the n^{th} block is asserted in Eq. (1).

$$H_n = Hash\Big(H_{n-1},\ PK_{L_n},\ Metadata_n,\ Transactions_n\Big) \tag{1}$$

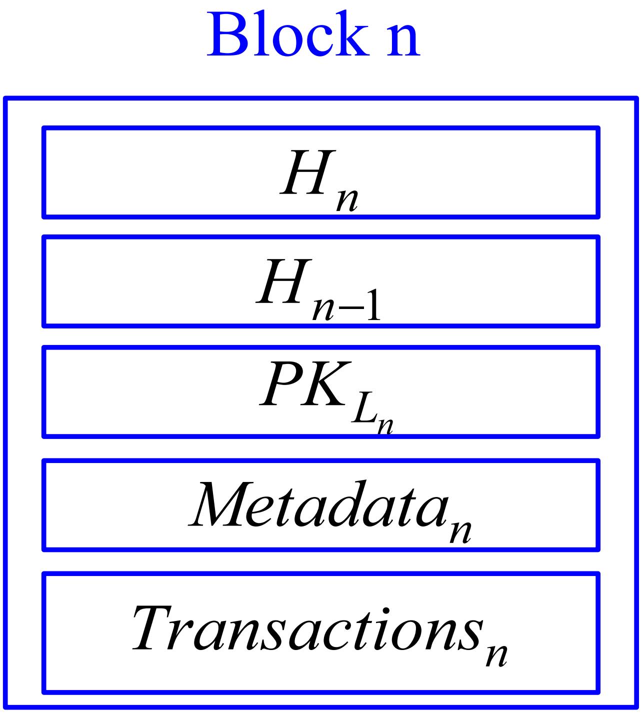


Fig. 3 The general structure of a block in a blockchain

4 Proof of humanity (PoH)

In this section, we outline the details of PoH. It is a leader selection consensus algorithm where a leader is selected from the pool of candidate leaders, i.e. the nodes that intend to store the block and attempt in the consensus, based on the contribution of the candidate leaders in donating to the organizations. Therefore, PoH potentially encourages the nodes to donate money to organizations. The participating nodes in PoH-based blockchain may generate smart contracts or/and transactions while the transaction fee is payable measured based on gas (inspired by Ethereum blockchain [16]). During bootstrapping, each organization creates a particular type of smart contract in the blockchain that contains a payable address and the reputation score for the organization. The payable address is the address that can be used by the participating nodes to donate to the organization, and the reputation score is an indicator of how much society trusts the organization. There are two functions defined for this specific kind of smart contracts: (i) Negative voting and (ii) positive voting. The participating nodes in the blockchain can call these functions in the smart contract setting to increase or decrease the organization's reputation score. The reputation score of the c^{th} organization is also called $Trust_c$. The trust increase or decrease of the c^{th} organization is formulated in (2). Voting in favor of or against an organization would cost a bit amount of money. If a huge number of society members participate in the reputation scoring process, no single entity can have a significant effect on the reputation score. Even if the entity makes a large number of accounts to vote, it is tremendously costly to neutralize the votes of the society member. To illustrate, consider that positive or negative voting costs 1 gas. If 10 million people vote in favor of an organization account, it costs 10 million gas to a single rival entity to neutralize these votes with negative votes.

$$Trust_c = \sum_{v \in V(c)} Vote_v \tag{2}$$

The more people vote in favor of or against organizations, the more the reputation score value reflects the public opinion. The number of society members who participate in the reputation scoring process is a sociological issue rather than a technical issue. This participation can be compared with political election participation. Evaluation of organization trust is an effective mechanism to supervise the honesty of the organizations. Such supervision brings about a new political era. The society member can vote for and against each organization which should spend money on helping poor communities, health care services, road construction, public education fund, etc. In what follows, the outline of the emerging political era required to practically implement this framework and, then, the mathematic behind the leader selection procedure in PoH are elaborated on.

4.1 Overview: General bureaucracy model

In the society-centric blockchain-based bureaucracy model, as depicted in Fig. 4, people pay their transaction fees to the leaders, and the leaders had previously paid a portion of it to the organizations. The more a candidate leader donates to the organization, the more it is likely to be a leader and take the incentive and transaction fees. Therefore, giving money to organizations resembles a competitive business for candidate leaders. Organizations are responsible for spending money on society's public expenditures. They should publicly declare the way they spend donations they received. Society gives reputation to them based on the declared spends; however, they might lie. Thus, the parliament supervises the accuracy of their claims. If there is a mismatch between the truth and declared spends, the parliament notices the society. Society gives a reputation to the organizations based on their public supervision and parliament warranted supervision. The more the reputation of an organization, the more likely is the candidate leaders who paid to the organization to be a leader. Thus, candidate leaders are willing to pay high reputation organizations. In short, the blockchain's leaders and organizations act as the financial sector of the administration in this model; except that they are continuously supervised and voted by society. If the society, at any time, finds any of them incapable of managing the tax, people can give a negative reputation to that organization. The way in which leaders are selected is discussed in the following section.

4.2 Leader selection algorithm

The participating nodes in the proposed blockchain pay to the payable address in the smart contract to donate to the organization. The amount of money donated by the transaction generator is used by the leader selection algorithm to identify the leader of the next block in the blockchain model. Mathematical implementation of the PoH is explained in detail next.

As previously mentioned, PoH selects the leader of the next block based on two factors: (i) the amount of money that is donated to the organization. A node may once donate to the organization and keep using the benefits to store new blocks. Note that the price of cryptocurrencies also fluctuates over time and, thus, some nodes may only donate when the price of the currency is low. To prevent this and to ensure a sustainable flow of donations, only denotations made in the latest N blocks are counted for the leader selection algorithm. The exact value of N depends on the application and can be

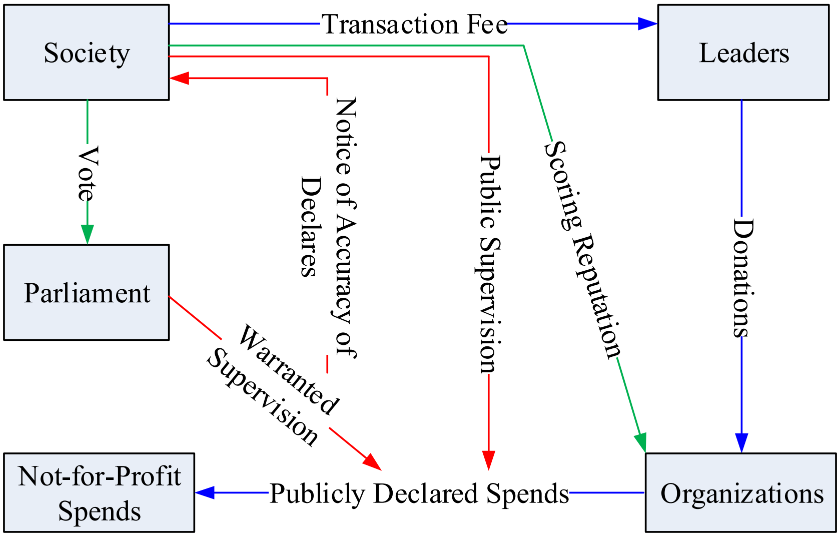


Fig. 4 Society-centric blockchain-based bureaucracy model

defined by the blockchain developers, and (ii) the reputation of the donation receiver organization.

The schematic of the leader selection procedure is summarized in Fig. 5. Each candidate account is assigned to an interval which has the length equal to the probability of becoming the leader, following which, a random number is generated. If the random number is within the assigned interval of account *a*, this account will become the leader. This stage of the process is inspired by the fundamental principles of scenario generation in the Monte-Carlo simulation [24].

Let $\mathbf{A} = \{a_1, a_2, \ldots, a_m\}$ be the set of addresses of candidate leaders, which are sorted from the smallest to the biggest value of the address. The probability of a candidate leader being selected as the leader is in accordance with Eq. (3).

$$PDF_a = DS_a = \frac{\sum Donation_{a,c}.Trust_c}{\sum\limits_{\forall a'} \sum Donation_{a',c}.Trust_c} \tag{3}$$

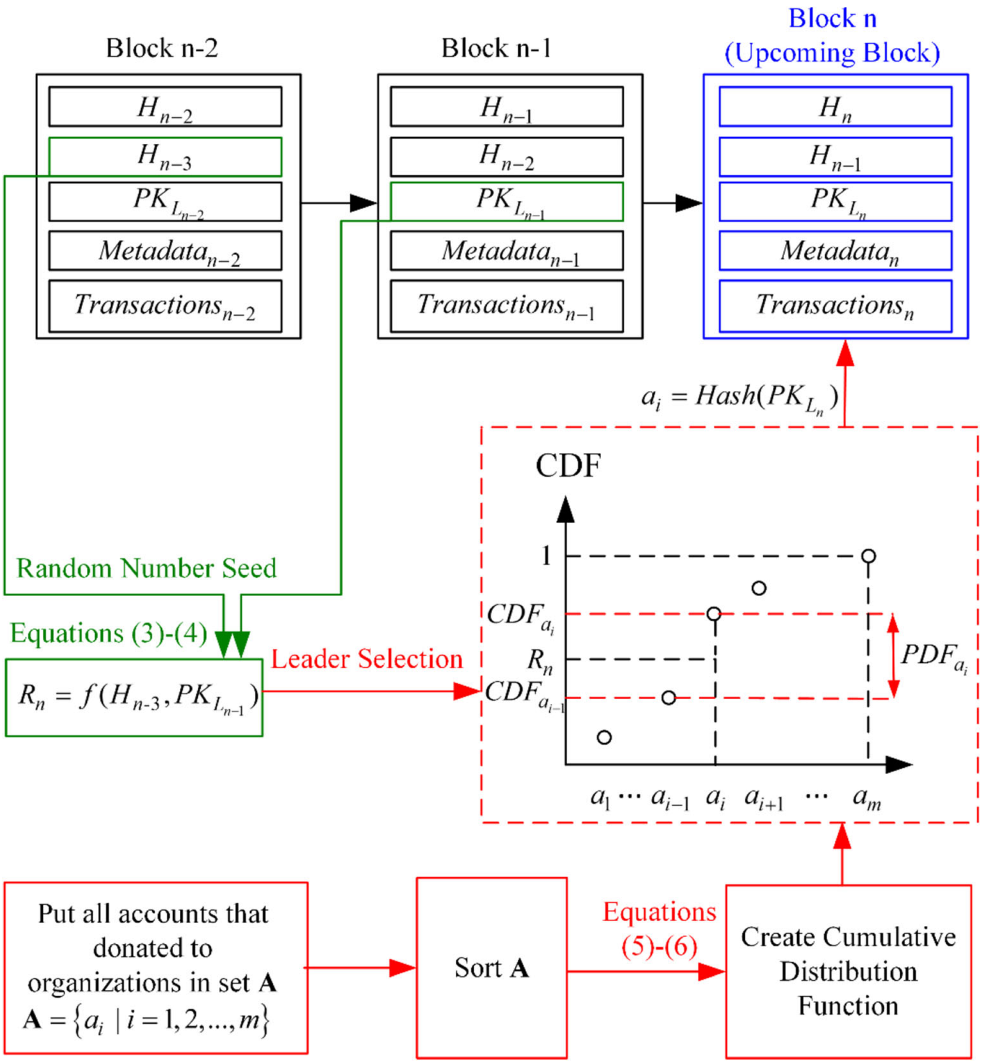
According to (3), the probability that the account *a* becomes the leader is equal to the total amount of its donations divided by the total donations received by all accounts. All donations are weighted by the trust (reputation) of the organization account receiving the donations.

Once DS_a is calculated, the nodes use (4) to assess the Cumulative Distribution Function (CDF). CDF is the summation of the probabilities that account *a* is selected as the leader plus the probabilities of the leadership of accounts whose addresses are less than or equal to *a* in value. The assigned interval to account a_i is $(CDF_{a_{i-1}}, CDF_{a_i}]$. The length of this interval is equal to PDF_{a_i}.

$$CDF_{a_i} = \sum_{a=a_1}^{a_i} PDF_a \tag{4}$$

Once the nodes evaluate (4), they start selecting a random leader. The participating nodes in the network calculate a

Fig. 5 The blockchain and PoH leader selection schematic



random variable R_n and compare the resulting value with CDF. The candidate leader a_i becomes the leader of the n^{th} block if the random number is within its assigned interval. As soon as the selected leader realizes this fact, he/she propagates the next block. Other nodes append the generated node to their instance of blockchain. Obviously, fork creation is impossible because once the block n-1 is propagated, all nodes compute the same values for donations shares, probabilities, and the generated random number. Therefore, they wait for the leader with the computed address. Generating the random variable (R_n) in a decentralized framework is associated with the following requirements:

1. The value of the random variable is not predictable by the participating nodes.
2. Participating nodes cannot influence the value of the random variable.

To fulfill the aforementioned requirements, a new method for generating random variables is developed in this paper. The random variable is presented in (5).

$$R_n = \frac{Hash\left(\text{Pub\_Key}_{L_{n-1}} + H_{n-3}\right)}{Maximum\_value\_of\_Hash} \tag{5}$$

The random function generator employs two fundamental values which are: (i) PK of the leader of the last block in the blockchain, and (ii) hash of the latest three blocks. The random function satisfies the first requirement outlined above as none of the participants knows the PK of the leader of the n-1*th* block before the propagation of this block, which in turn, makes it impossible to predict R_n. The second requirement is also satisfied as the leader of the n-1*th* block cannot change its PK, thus he/she has no control over R_n. Since no one is capable of adjusting the random number, leader selection manipulation is impossible as well.

In case the new leader does not propagate the block within a pre-defined time interval known as *creation deadline* (*CD*), a new leader is selected through the following random number generating equation:

$$R_n^{'} = \frac{Hash\left(Hash\left(\text{Pub\_Key}_{L_{n-1}} + H_{n-3}\right)\right)}{Maximum\_value\_of\_Hash} \tag{6}$$

5 Resilience issue

As explained in Section 3, there is no possibility for fork creation. Therefore, the double-spending attack, which is related to fork creation, is no longer a threat. In consequence, the chain would certainly become immutable. In addition, if a leader propagates a false block, the other nodes refuse to add it to the blockchain. Hence, the leader does not add false information inside the block. An invalid block may contain a transaction that contrasts transactions of previous blocks or an unauthentic transaction in which the signature does not consist with the public key of the transactor. The only threat is propagation prohibition. According to this threat, some nodes refuse to propagate the block they received from a neighbor node to the other nodes. This behavior not only limits the speed of the blockchain network but also results in some nodes not receiving a generated block within the creation deadline. We categorize nodes into three groups: (1) *normal nodes* which act as expected, (2) *malicious nodes* which deliberately refuse to propagate the block resulting in the above concern, (3) *lazy nodes* which, with no intention, do not pay attention to the propagation tasks. In the following subsections, it will be discussed how the network reconfiguration can enhance the resilience of the network against this threat.

5.1 Network reconfiguration

In a peer-to-peer blockchain network, there is no central server. A node should establish connections with some nodes via a specific port. Connection protocol is almost similar to the Bitcoin network. A node connects to another node by the handshaking process [25]. In this regard, he/she sends a version message (with JSON format) to the *IP: Port_of_this_Blockchain*. The receiver can accept the connection by responding to the version message with his/her wallet version massage.

Once a node is connected to the network, it can get the *IPs* of the other participants with *getaddress* message. A node which receives this message will respond to this request by sending the set of all *IPs* that have been recognized in the network.

Due to the resilience concern discussed earlier, a node regularly changes its connected peers. This procedure is called *reconfiguration.* This paper suggests that reconfiguration be accomplished once in a time period called *reconfiguration duration.* A node tends to establish almost three connections. In this regard, each node may request another node as a new connection with the probability of 3 divided by Number_of_Nodes. Since other nodes may also request the node as a connection, 3 + 3 nodes are connected to the node on average. Changing the connection to other nodes provides the nodes with more probability of receiving the new blocks.

6 Evaluations and discussions

In this section, we study the performance of PoH through simulations. We used Matlab [26] to simulate PoH and studied the performance on a PC with Intel core i5 CPU and 6 GB

RAM. In the following subsections, metrics of resilience, the average block creation duration, and computational overhead are evaluated. Furthermore, a comparison of the PoH and other known consensus algorithms are discussed.

6.1 Resilience evaluations and discussions

In order to investigate the resilience of the proposed blockchain, mobility of blocks, which refers to the propagation speed in the blockchain network, is simulated. The simulation horizon is 20 min, and 500 nodes participate in the blockchain network. Each normal node passes the most recent block to its neighbors whenever it assures the validity of the block. The validation includes three stages: (i) checking the transaction validation, (ii) checking whether header and metadata are built correctly, and (iii) checking whether the hash of the public key of the block creator is consistent with the leader selection protocol.

We make the following assumptions:

1. A malicious or lazy node hesitates to propagate a valid block to the rest of the network. In contrast, a normal node sends the valid block to its neighbor nodes.
2. It takes 2 s to check the validity of a block.

In order to investigate the mobility of a valid block into the network, as an adverse situation, only 40% of the nodes are considered normal. Ten thousand different simulation instants are simulated. Two cases are studied accordingly:

Case 1: The network configuration is constant. This case is analyzed to demonstrate that constant network configurations might endanger the resilience of the blockchain.
Case 2: A node will alter its neighbors every 60 s.

The results of these two cases are presented in Figs. 6 and 7. In these figures, the number of nodes that received the block is depicted as the number of aware nodes. According to Fig. 6, the expected number of nodes that receive the block after 20 min in Case 1 is below 500. Therefore, some nodes might not receive the block and wait for a new block from a new leader. The reason lies mainly in the fact that malicious or lazy nodes block the path to deliver valid blocks to some normal nodes. In contrast, as depicted in Fig. 7, the expected number of nodes which receive the block is 500. Thus, reconfiguring the blockchain network is critical in order to keep the blockchain secure. Since reconfiguration provides the opportunity to create new paths to deliver valid blocks to normal nodes, the network resilience enhances significantly. The percentage of scenarios in which all nodes receive the block within a specified time is shown in Fig. 8. This figure depicts that in all the 10,000 scenarios, the block is propagated within 800 s. The result reveals that by reconfiguring the connection of nodes in a proper period, all nodes will receive valid blocks within 20 min even if the normal nodes are the minority.

In another case study, propagation of a block through a network with various percentage of normal nodes and various reconfiguration periods are simulated for 10,000 scenarios on a network of 500 nodes. The results are shown in Figs. 9 and 10. These figures reveal that the more the percentage of the normal nodes is, the sooner all nodes receive the block. Besides, a smaller duration of reconfiguration leads to faster propagation. As Shown in Fig. 9, two reconfiguration actions are sufficient for a network with 40% normal nodes. However, as the percentage of normal nodes decreases, additional reconfiguration actions would be required, as depicted in Fig. 10.

6.2 Block creation duration

After the leader of a new block receives the previous block, he/she calculates the donation shares and random number based on (3)–(6), and then realizes that he/she is selected as the leader. He/she immediately releases the new block. Therefore, the block creation duration is equal to the time a leader receives the previous block. However, there is an exception. When the leader is inactive, other nodes wait until the end of the creation deadline, which is 20 min in this case.

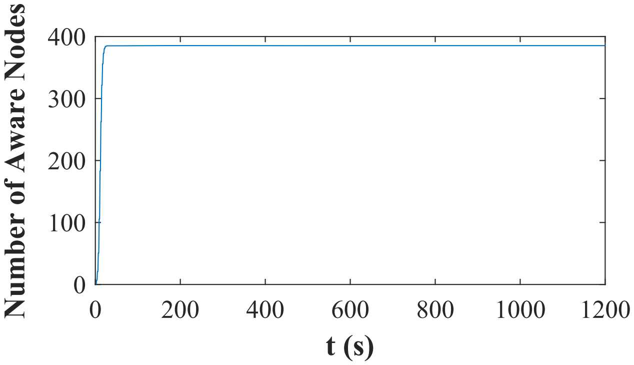


Fig. 6 Mobility of a valid block through the blockchain network in Case 1

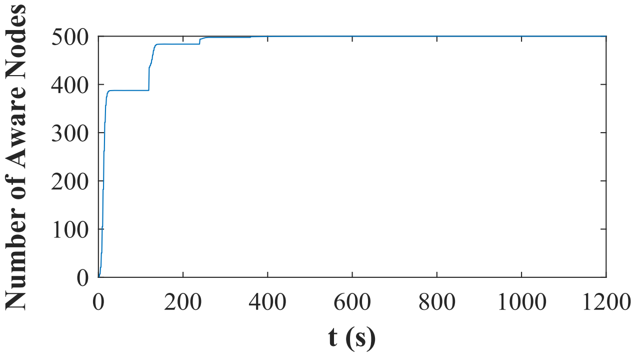


Fig. 7 Mobility of a valid block through the blockchain network in Case 2

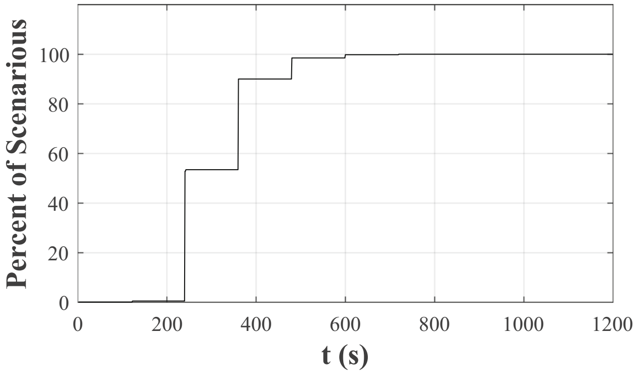


Fig. 8 Percent of scenarios in which all nodes receive the valid block within a specific time in Case 2

In order to compute the average creation time, the creation of 8000 blocks is simulated. It is assumed that 2 % of nodes are either malicious or lazy, and 1 % of the selected leaders are inactive. Nodes change their neighbors every 40 s. As a conservative assumption, it is also assumed that leader of the new block is the last node that receives the previous block. As depicted in Fig. 11, the average block creation duration is 46 s if there are 500 nodes in the network. As the number of nodes increases, the average creation duration increases as well. Nevertheless, this change is insignificant, and thereby not a considerable concern.

6.3 Computational overhead

Another evaluation metric of a blockchain that has been investigated in the literature is the computation overhead [27]. In this section, the computation overhead of the proposed PoH consensus algorithm is analyzed via investigating the computational complexity of each element of the consensus process. Then, the most significant complexity would reflect the complexity of the entire consensus algorithm.

At the first stage of this algorithm, the m candidate leaders are sorted based on their addresses. *Merge-Sort* algorithm, which has the complexity of $O(m\log m)$ [28], is used in this stage of the consensus algorithm. The probability distribution density and cumulative distribution function are computed once for each candidate leader. Therefore, these computations are $O(m)$ complex. Generating a random number computation burden is not affected by the number of candidate leaders, thereby being of $O(1)$ complexity. The generated random number is compared with each candidate leaders' cumulative distribution function in order to select a leader. This stage has the complexity of $O(m)$. Putting all together, the PoH consensus algorithm has the complexity of $O(m\log m)$. In order to control the complexity of the algorithm, m should be kept small or medium. Recall, from Subsection 4.2, that only the accounts who have donated in the N previous blocks are candidate leaders. Therefore, by decreasing N properly, m would be smaller, and thereby computation overhead of the consensus algorithm is lowered.

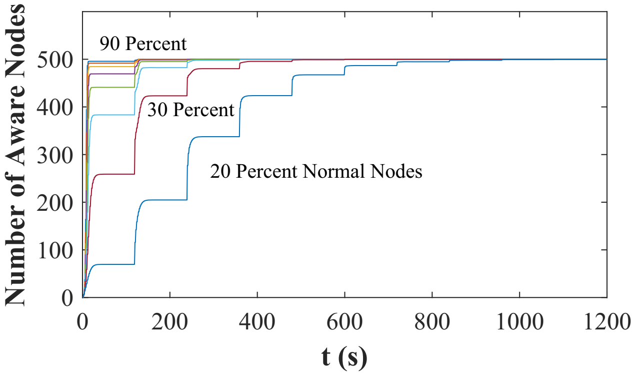


Fig. 10 Mobility of a valid block through the blockchain network in different percentage of normal nodes considering 120 s reconfiguration duration

6.4 Comparison with existing consensus algorithms

A qualitative comparison of the PoH consensus algorithm with some of the most known existing consensus frameworks is provided in Table 1. The existing consensus algorithms were discussed in Section 2 in detail. Proof of Humanity's

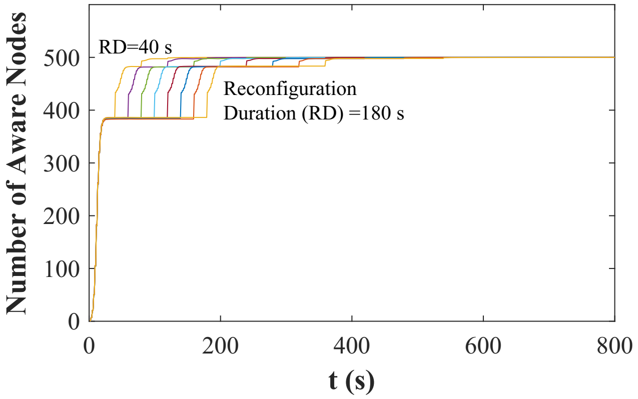


Fig. 9 Mobility of a valid block through the blockchain network in different reconfiguration duration considering 40% normal nodes

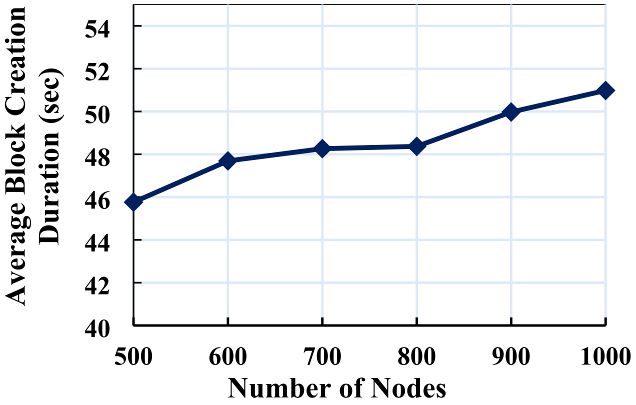


Fig. 11 Average simulated duration of a new block creation in the proposed blockchain

Table 1 Comparison of the proposed PoH consensus and state-of-the-art consensus algorithms

Consensus Algorithm	PoW	PoB	PoS	PoA	PoH
Robust Against Fork Creation	*N	*Y	*S	Y	Y
Energy Cons. Efficient	N	Y	Y	Y	Y
Decentr.	Y	Y	Y	N	Y
Tax Gathering Enabled	N	N	N	N	Y

*Y: Yes

*N: No

*S: In some versions

main features is also introduced in Section 4. As summarized in Table 1, the Proof of Humanity is the only blockchain consensus algorithm that enables gathering society-related taxes while is energy-consumption efficient, robust against fork creation, and managed in a fully decentralized manner.

7 Conclusion

Motivated by the societal concerns and the need for reducing the huge amount of energy consumptions, this paper proposed the Proof of Humanity (PoH) consensus framework. In this framework, the more a candidate leader donates to a trusted non-profit organizations, the more he/she is likely to be the leader of the upcoming blocks. Since the value of society-related taxes is not regulated, the competition dynamic of leaders determines the amount of donation. In addition, the general structure of the society-centric blockchain-based bureaucracy is introduced. The way people supervise the performance of the organizations also discussed. Practical implementation of the PoH needs a mechanism for a decentralized truly random variable generation. This paper suggested a random variable generation algorithm that suits for blockchains' decentralized architectures. The generated random number is neither predictable nor influenced by a group of entities. Simulations demonstrated that the proposed blockchain designed based on the PoH concept in this paper is highly resilient even in some adverse scenarios in which the normal nodes are in minority. To maintain the resilience, nodes should regularly change their neighbor nodes that they are connected to. The more frequently they change their connection, the more resilient the network is, and valid blocks propagate more promptly. The average block creation duration is also evaluated through simulations. It has been observed that a block is added to the network between 46 to 51 s for a network of 500 to 1000 participant nodes, which indicates a fast blockchain network.

The proposed framework is suitable to be deployed as the main financial cryptocurrency system in the countries. While this paper focused on the general framework of society-centric blockchain and required consensus algorithm; detailed technical design, programming, and practical test are suggested for future direction of research.

Nomenclature

Indices and sets *a*, Index of candidate leader account; *c*, Index of organization account; *n*, Index of blocks; V(*c*), Set of all votes to organization *c*; *v*, Index of votes

Variables and functions **B**, Blockchain; CDF_a, Cumulative distribution function that the account *a* is selected as a leader, i.e. the probability that the selected leader has the address equal or less than *a* in value; $Donation_{a,c}$, Donation of account *a* to organization account *c*; DS_a, Donations share of candidate leader account *a*; H_n, Header of block *n*; *Hash*(.), Hash function; L_n, Leader of block *n*; PDF_a, Probability distribution function that the account *a* is selected as a leader; $Pub\_key_{L_n}$, The public key of the leader of n^{th} block; R_n, Random number used for selecting a leader for n^{th} block; $Trust_c$, The variable which indicates the trust of people to organization account *c*; $Vote_v$, The vote *v* (1 if positive vote and −1 if negative vote)

References

1. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Bitcoin. https://bitcoin.org/bitcoin
2. Underwood S (2016) Blockchain beyond bitcoin. Communications of the ACM 59(11):15–17
3. Bashir I (2017) Mastering blockchain. Packt Publishing Ltd, Birmingham
4. Jun M (2018) Blockchain government-a next form of infrastructure for the twenty-first century. J Open Innov: Technol Market Complexity 4(1):7
5. Esposito C, Ficco M, Gupta BB (2021) Blockchain-based authentication and authorization for smart city applications. Inf Process Manag 58(2):102468
6. Dorri A, Kanhere SS, Jurdak R, Gauravaram P (2019) LSB: a lightweight scalable Blockchain for IoT security and anonymity. J Parallel Distributed Comput 134:180–197
7. Gupta BB, Quamara M (2020) An overview of internet of things (IoT): architectural aspects, challenges, and protocols. Concurrency Comput: Pract Exp 32(21):e4946
8. Fan Y, Zou J, Liu S, Yin Q, Guan X, Yuan X, Wu W, Du D (2020) A blockchain-based data storage framework: a rotating multiple random masters and error-correcting approach. Peer-to-Peer Network Appl 13(5):1486–1504
9. Khalid ZM, Askar S (2021) Resistant Blockchain cryptography to quantum computing attacks. Int J Sci Business 5(3):116–125
10. Gupta BB (2020) An efficient KP design framework of attribute-based searchable encryption for user level revocation in cloud. Concurrency Comput: Pract Exp 32(18):e5291
11. Gupta B, Agrawal DP, Yamaguchi S (eds) (2016) Handbook of research on modern cryptographic solutions for computer and cyber security. IGI global. Hershey PA, USA
12. Yu C, Li J, Li X, Ren X, Gupta BB (2018) Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. Multimed Tools Appl 77(4):4585–4608
13. M. Castro, Liskov (1999) Practical Byzantine fault tolerance." In OSDI, vol. 99, no. 1999, pp. 173–186

14. Li Y, Qiao L, Lv Z (2021) An optimized byzantine fault tolerance algorithm for consortium Blockchain. Peer-to-Peer Network Appl 16:1–4
15. Asayag A, Cohen G, Grayevsky I, Leshkowitz M, Rottenstreich O, Tamari R, Yakira D (2018) Helix: a scalable and fair consensus algorithm resistant to ordering manipulation. IACR Cryptol ePrint Arch 2018:863
16. Buterin V (2014) Ethereum white paper: a next generation smart contract & decentralized application platform. White Paper 3:37
17. De Angelis S, Aniello L, Baldoni R, Lombardi F, Margheri A, Sassone V (2018) PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain. Italian Conference on Cyber Security, Milan, Italy
18. What is proof of burn (eli5)?, Slimcoin. https://slimcoin/proof-of-burn-eli5/
19. Antonopoulos AM, Wood G (2018) Mastering Ethereum: building smart contracts and dapps. O'Reilly Media. Newton, Massachusetts, United States
20. Nguyen CT, Hoang DT, Nguyen DN, Niyato D, Nguyen HT, Dutkiewicz E (2019) Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. IEEE Access 7:85727–85745
21. Bentov I, Gabizon A, Mizrahi A (2016) Cryptocurrencies without proof of work. In: International Conference on Financial Cryptography and Data Security. Springer, Berlin, pp 142–157
22. Sun G, Dai M, Sun J, Yu H (2020) Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain. IEEE Internet Things J 8(8):6257–6272
23. Liu W, Li Y, Wang X, Peng Y, She W, Tian Z (2021) A donation tracing blockchain model using improved DPoS consensus algorithm. Peer-to-Peer Network Appl 10:1–2
24. Billinton R, Allan RN (1992) Reliability evaluation of engineering systems- concepts and techniques(book). Plenum Press, New York, USA
25. Antonopoulos AM (2014) Mastering Bitcoin: unlocking digital cryptocurrencies. O'Reilly Media. Newton, Massachusetts, United States
26. MATLAB (Programming Language) (2017) The MathWorks, Natick, MA, USA
27. Zhang Y, Xu C, Cheng N, Li H, Yang H, Shen X (2019) Chronos+: an accurate Blockchain-based time-stamping scheme for cloud storage. IEEE Trans Serv Comput 13(2):216–229
28. Goodrich MT (2014) Roberto Tamassia, Michael H. Goldwasser. Data structures and algorithms in Java. John Wiley & Sons. Hoboken, NJ, USA

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ali Arjomandi-Nezhad received the B.Sc. degree in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2016, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2018. He is currently a research assistant at the Sharif University of Technology. One of his honors is taking the first rank of Electrical Engineering Olympiad of Iran in 2016. His research interests include both computer science and power systems engineering. Azadi Ave., P.O.Box 11155-4363, Tehran, Iran. a.arjomandi@alum.sharif.edu

Mahmud Fotuhi-Firuzabad (IEEE Fellow, 2014) Obtained B.Sc. and M.Sc. Degrees in Electrical Engineering from Sharif University of Technology and Tehran University in 1986 and 1989 respectively and M.Sc. and Ph.D. Degrees in Electrical Engineering from the University of Saskatchewan, Canada, in 1993 and 1997 respectively. He is a professor of Electrical Engineering Department and president of Sharif University of Technology, Tehran, Iran. He is a member of center of excellence in power system control and management in the same department. His research interests include power system reliability, distributed renewable generation, demand response and smart grids. He is the receipient of several national and international awards including Sixteen Khwarizmi International award, World Intellectual Property Organization (WIPO) award for the outstanding inventor, 2003, PMAPS International Society Merit Award for contributions of probabilistic methods applied to power Systems in 2016 and 2014 Allameh Tabatabaei award. Dr. Fotuhi-Firuzabad is a visiting professor at Aalto University, Finland. He serves as the Editor-In-Chief of the IEEE POWER ENGINEERING LETTERS. He is a Fellow of the IEEE for his contribution to to the application of probabilistic techniques in power system reliability assessment. Azadi Ave., P.O.Box 11155-4363, Tehran, Iran. fotuhi@sharif.edu

Ali Dorri is a Research Fellow at Queensland University of Technology (QUT), Brisbane, Australia. He received his Ph.D. degree from the University of New South Wales (UNSW), Sydney, Australia. He was also a Postgraduate research student at CSIRO, Australia. His core publications on blockchain for IoT have received tremendous attention and one of his papers is continuously ranked among the most downloaded conference papers in IEEE explorer (top 50 and in some months the second rank). Two of his papers are top-cited in their respective venues. His publications are cited over 2500 times and Ali has h-index of 15. His research interest includes blockchain, Internet of Things (IoT), security and privacy, and distributed systems. He has published over 30 peer-reviewed paper and an authored book titled "blockchain for cyber physical systems". Ali served on the organizing committee of multiple conferences including ICBC and BCCA. 2 George St, Brisbane City QLD 4000, Australia. ali.dorri@qut.edu.au

Payman Dehghanian (S11, M17, SM20) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2009, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2011, and the Ph.D. degree in electrical engineering from Texas A&M University, Texas, USA, in 2017. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The George Washington University, Washington, DC, USA. His research interests include power system protection and control, power system reliability and resiliency, asset management, and smart electricity grid applications. Dr. Dehghanian was a recipient of the 2013 IEEE Iran Section Best M.Sc. Thesis Award in Electrical Engineering, the 2014 and 2015 IEEE Region5 Outstanding Professional Achievement Awards, and the 2015 IEEE-HKN Outstanding Young Professional Award. 800 22nd St. NW, Washington, DC 20052, USA. payman@email.gwu.edu