Public Participation Consortium Blockchain for Smart City Governance

Yuhao Bai, *Graduate Student Member, IEEE*, Qin Hu, Seung-Hyun Seo, *Member, IEEE*, Kyubyung Kang, *Associate Member, IEEE*, and John J. Lee, *Senior Member, IEEE*

***Abstract*—Smart cities have become a trend with improved efficiency, resilience, and sustainability, providing citizens with high quality of life. With the increasing demand for a more participatory and bottom–up governance approach, citizens play an active role in the process of policy making, revolutionizing the management of smart cities. In the example of urban infrastructure maintenance, the public participation demand is more remarkable as the infrastructure condition is closely related to their daily life. Although blockchain has been widely explored to benefit data collection and processing in smart city governance, public engagement remains a challenge. In this article, we propose a novel public participation consortium blockchain system for infrastructure maintenance that is expected to encourage citizens to actively participate in the decision-making process and enable them to witness all administrative procedures in a real-time manner. To that aim, we introduced a hybrid blockchain architecture to involve a verifier group, which is randomly and dynamically selected from the public citizens, to verify the transaction. In particular, we devised a private-prior peer-prediction-based truthful verification mechanism to tackle the collusion attacks from public verifiers. Then, we specified a Stackelberg-game-based incentive mechanism for encouraging public participation. Finally, we conducted extensive simulations to reveal the properties and performances of our proposed blockchain system, which indicates its superiority over other variations.**

***Index Terms*—Blockchain, consensus algorithm, smart city, smart governance.**

I. Introduction

DUE TO the rapidly increasing density of population in urban areas, citizens' quality of life has been heavily affected by environmental resource constraints, traffic congestion, air pollution, greenhouse gas emission, and waste disposal [1]–[3]. All these challenges and problems require main participants in cities, i.e., governments and citizens, to pay attention to utilize technologies and innovations to enhance the efficiency, resilience, and sustainability of urban systems, thus continuously benefiting citizens' quality of life.

Urban infrastructure is a complex sociotechnical system, which affects citizens' daily life and even the future development of the city. Only if infrastructure initiatives are planned, operated, and maintained correctly can they have a positive impact on the wellbeing and growth of society [4], [5]. As demonstrated in [6], the governance approach plays a key role in leading such initiatives to success or failure. In the past few decades, conventional governance methods have achieved great success in providing better services to more citizens. While with the help of advanced information and communication technology, the transition toward smart cities has begun, which also brings a character to the urban governance, *public engagement*. Inspired by new channels of communication and expression of opinions, such as social networks and mobile technologies, citizens have increasing demand to participate in the decision-making process, which challenges the traditional top–down paradigm of governance and agenda setting. Thus, it is essential to design a system embracing the citizens' voices and opinions as a continuous input into the decision-making process and involving citizens in real-time monitoring of public administrations.

Due to the attractive characteristics of decentralization, transparency, and immutability, blockchain is considered as a potential approach to promoting public engagement in urban governance to achieve trustless collaboration among multiple parties [7], [8]. Some works deploy blockchain among authoritative organizations to promote information sharing and cooperation.

For example, various works [9]–[13] have studied the application of blockchain technology in supply chain management, where the blockchain was utilized to track the detailed information of products and share business information among entities in supply chains. However, in this scenario, information is conveyed from the governmental or regulatory

Manuscript received March 17, 2021; revised May 31, 2021; accepted June 15, 2021. Date of publication June 22, 2021; date of current version January 24, 2022. This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government (MSIT) under Grant 2020-0-01568, and in part by the National Research Foundation of Korea through the Korean Government under Grant 2018R1A2B6006903. *(Corresponding authors: Qin Hu; Seung-Hyun Seo.)*

Yuhao Bai is with the Department of Electronic and Electrical Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: byh2018@hanyang.ac.kr).

Qin Hu is with the Department of Computer and Information Science, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: qinhu@iu.edu).

Seung-Hyun Seo is with the Division of Electrical Engineering, Hanyang University, ERICA Campus, Ansan 15588, South Korea (e-mail: seosh77@hanyang.ac.kr).

Kyubyung Kang is with the Department of Engineering Technology, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: kyukang@iu.edu).

John J. Lee is with the Department of Electrical and Computer Engineering, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: johnlee@iu.edu).

Digital Object Identifier 10.1109/JIOT.2021.3091151

agency of the initiative to the public. The public can access and use the shared business information, but public feedback is not required. On the other hand, some works utilized blockchain to benefit applications requiring interaction between the public and institutions, such as governments or organizations. As an example, Yavuz *et al.* [14] introduced an e-voting system, where the vote records are stored on the Ethereum blockchain. In this case, information is transferred from the public to the governmental agency, following a process initiated by the agency. In particular, there is no formal communication channel between individual members of the public and the agency.

According to Rowe and Frewer [15] three-level public engagement model, the existing blockchain-based smart governance approaches mainly focus on promoting the first two levels of engagement, i.e., *public communication* and *public consultation*, where the first level is top–down information flow, and the second level collects the opinions of the public. However, due to the lack of mechanisms to provide two-way dialogue channels for the public and the authorities, the existing works could not achieve the third-level public engagement, *public participation*, which involves the information exchange between the public and the government.

In this article, we propose a *public participation* consortium blockchain system for smart infrastructure maintenance. First, in order to involve the citizens in decision-making processes, we introduce a hybrid blockchain architecture, where a verifier group, which is randomly and dynamically selected from the public, is employed in the transaction verification. Then, we propose a new consensus algorithm for our blockchain system, which can deal with an additional verifier group since existing consensus protocols do not have that ability. We propose a private-prior peer-prediction-based truthful verification mechanism to resist collusion attacks from the public verifiers. Unlike the classical peer prediction method, our approach relaxes the requirement of common prior belief amongst both the public verifiers and the mechanism. In our approach, each verifier could keep a subjective and private belief. In addition, a Stackelberg-game-based incentive mechanism is devised to elicit public citizens' active and continuous participation in verification.

In a nutshell, the main contributions of this article are as follows.

1) To the best of our knowledge, the proposed public participation consortium blockchain-based infrastructure maintenance system is the first smart governance approach with the citizens' voices and opinions included in the decision-making real-time monitoring of public administration. Compared to the traditional consortium blockchain, we involve a public verifier group in the consensus process for transaction verification.
2) We develop a private-prior peer-prediction-based truthful verification mechanism to reach consensus in a secure manner, where the reports of public verifiers are evaluated and inferred based on a trustworthiness model to eliminate the negative impacts of unreliable reports and motivate verifiers to report truthfully.
3) We devise a Stackelberg-game-based incentive mechanism for encouraging continuous public participation in the transaction verification process.
4) Extensive simulations are conducted to analyze the properties and performances of the proposed public participation consortium blockchain.

The remainder of this article is organized as follows. Most related works are reviewed in Section II. The system model is introduced in Section III, and the detailed design of the consensus algorithm is presented in Section IV. Experimental evaluation results are reported in Section V. Section VI concludes the whole article.

II. Related Work

A. Blockchain-Based Smart Governance

A modern city is an enormously complex ecosystem composed of stakeholders from different fields, e.g., local governments, citizens, and corporations, with conflicting interests. The transition toward smart cities introduces new challenges to the classical city governance practices. Meanwhile, the complexity of the urban infrastructure makes the problem more severe since no single party has the necessary knowledge, expertise, power, or resources to tackle these challenges alone [6], [16], [17]. As a result, a smart governance system is required to coordinate all forces and ensures that decisions are appropriately made, effectively implemented, and carefully evaluated [18]. Via applying blockchain to build an efficient smart governance system, the existing studies can be roughly divided into two categories: 1) authority-oriented and 2) public-oriented, where the former type of work deploys blockchain among authoritative organizations to promote information sharing and cooperation. At the same time, the latter utilizes blockchain to involve public engagement.

Authority-oriented blockchain-based smart governance covers particular professional fields, such as architectural design, supply chain management, and public healthcare [9]–[13], where the system efficiency is restricted by insecure, untimely communication or uncooperative work. Dounas *et al.* [19] developed a framework for a decentralized architectural design using building information modeling (BIM) agents connected using Ethereum. By integrating BIM with blockchain, this framework scales collaboration to thousands of agents and shifts trust to the infrastructure rather than the architectural design team. Liao and Wang [20] studied the application of blockchain in integrated casinos and entertainment (ICE) logistics, which improves the efficiency of logistics while ensuring the privacy of transactions. Azaria *et al.* [21] introduced a blockchain-based record management system, named *MedRec*, to handle decentralized electronic medical records, which addresses the issue of fragmented and slow access to medical data while improving the data quality and quantity for medical research.

Public-oriented smart governance benefits applications requiring interaction between the public and authoritative organizations, including smart grid, e-voting, intelligent transportation system, data storage, and sharing [22]–[26]. Liu *et al.* [27]

introduced an adaptive blockchain-based electric vehicle participation (AdBEV) scheme to minimize the power fluctuation level in the grid network and the overall charging cost for EV users, in which an iceberg order execution algorithm is applied to obtain an improved EV charging and discharging schedule. Yang *et al.* [28] proposed a decentralized trust management system in vehicular networks, based on which vehicles can validate the received messages from neighboring vehicles with the trust value being calculated and managed by RSUs. Shafagh *et al.* [29] presented a blockchain-based auditable data management system for IoT, which allows for fine-grained access control and sharing of time-series sensor data for various IoT applications. Yavuz *et al.* [14] implemented an e-voting application as a smart contract on Ethereum to guarantee the integrity, authenticity, and nonrepudiation of the vote records. To achieve voting on community projects, *Coinstack* [30] is developed based on Bitcoin by the province of Gyeonggi-do in South Korea and proved to be compatible with Ethereum smart contracts. Besides, *Polys* [31] is an Ethereum smart-contract-based e-voting system, which was launched in November 2017 by applying a distributed ledger technology (DLT) system. To ensure the anonymity of votes, voting calculations are encrypted.

In general, the existing blockchain-based smart governance mainly focuses on the first two levels of public engagement but provides no two-way communication channels for public groups and authoritative organizations. However, as infrastructure maintenance requires dialogue between citizens and government, the studies mentioned above are not suitable for this scenario.

B. Consensus Protocols of Blockchain

The existing consensus protocols are mainly applicable to permissionless and permissioned blockchains. In Table I, all significant consensus protocols mentioned in the following are summarized from four aspects, i.e., throughput, latency, adversary tolerance, and overhead cost.

In the permissionless blockchain consensus, the most widely known protocol is the Proof of Work (PoW) [32], which has proved to be an effective approach for cryptocurrencies over the years. As we all know, it suffers from high computational and bandwidth consumption. Proof of Capacity (PoC) [33] is a similar concept while it consumes disk space rather than computing resources to mine a block, which is an energy-efficient protocol compared to PoW. Proof of Stake (PoS) [34] is the second-most prevalent consensus used for cryptocurrencies. To determine the next block, instead of demanding users to find a nonce in unlimited space, PoS requires people to prove the ownership of the amount of currency. PoS does not consume high computational power. Delegated PoS (DPoS) [35] is the most typical variation of PoS, which is a representative democratic approach with stakeholders voting to choose some nodes as validators. Proof of Activity (PoA) [36] appears, where miners are elected according to PoW to generate a block, then the new block is signed by a group of validators selected using PoS. PoA combines the benefits of POW with that of POS. The stellar consensus protocol (SCP) [37], developed based on the federated Byzantine fault tolerance (FBFT), is the first Byzantine-agreement-based consensus method, which provides users with the maximum freedom to choose among different combinations of other participants to trust in order to reach a consensus. Similar to Stellar, Ripple [39] uses FBFT, where two types of nodes are defined, i.e., server nodes responsible for the consensus protocol and client nodes joining via transferring funds. Tangle [38] is based on the directed acyclic graph (DAG), where each transaction is a unique block by itself and has to approve two older transactions to be included.

In the consortium blockchain consensus protocols, the prominent one is the practical Byzantine fault tolerance (PBFT) [40], where all nodes participate in the voting process to add the following block, and the consensus is reached when more than 2/3 nodes agree upon that block. Delegated Byzantine fault tolerance (dBFT) [41] follows the same rule but does not require the participation of all nodes, where some nodes are chosen as delegates of other nodes. Tendermint [42] is a hybrid consensus protocol based on PBFT and PoS, where nodes have different voting powers proportional to their stakes. Proof of Elapsed Time (PoET) [43] works similar to PoW but consumes significantly less energy. In PoET, miners have to solve a hash problem similar to that in PoW, but the winning miner is randomly chosen based on a random wait time. The verification of correctness of timer execution is done using a trusted execution environment (TEE). Raft [44] is a voting-based consensus protocol, which is designed to make Paxos algorithm [45] more understandable and implementable for practical systems and composed of two stages: 1) leader election and 2) log replication. The leader is responsible for ordering the transactions, after which the leader accepts log entries from clients and broadcasts transactions to make its version of the transaction log. In our system, the Raft algorithm is deployed to elect a leader from the consortium network. This "leader" is elected from the predefined consortium group, which consists of trustworthy nodes selected by the authorities of infrastructure maintenance. We assume that the leader is reputable because the authority filters out unreputable candidates in the predefined consortium group. This guarantees that the leader reaches a certain standard for its reputation. Thus, we chose not to deploy a separate "reputation" mechanism in our system. However, the reputation scheme is absolutely an important aspect in leader election, especially when electing the leader in an untrustful public environment. In peer-to-peer networks, reputation systems are used to drive mutual trust among participants and promote successful interaction [46]. There are many works adopting reputation mechanisms to elect a leader in the consensus process for public environments. Zhuang *et al.* [47] introduced a reputation-based consensus protocol called proof of Reputation (PoR). Nodes in PoR build their high reputation by actively participating in system transaction consensus to gain cooperation from other nodes, and the new block is generated by the leader node with the highest reputation. In RepuCoin [48], a reputation-based weighted voting consensus is employed, where a miner's decision power is given by its reputation. Consensus is carried out by a group of miners with top reputations, and every member in this

TABLE I
COMPARISON OF DIFFERENT CONSENSUS PROTOCOLS

	Consensus	Throughput	Latency	Adversary Tolerance	Overhead		
					Computing	Network	Storage
Permissionless	PoW [32]	low	high	25% computing power	high	low	high
	PoC [33]	low	high	-	low	low	extremely high
	PoS [34]	low	medium	50% stakes	medium	low	high
	DPoS [35]	high	medium	51% validators	medium	-	high
	PoA [36]	low	medium	51% online stakes	high	low	high
	SCP [37]	high	medium	variable	low	medium	high
	Tangle [38]	high	low	33% computing power	low	low	low
	Ripple [39]	high	medium	20% faulty UNL nodes	low	medium	high
Permissioned	PBFT [40]	high	low	33% faulty replicas	low	high	high
	dBFT [41]	high	medium	33% faulty replicas	low	high	high
	Tendermint [42]	high	low	33% voting power	low	high	high
	PoET [43]	high	low	-	low	low	high
	Raft [44]	high	low	50% crash fault	low	-	high

group has a weight associated to its vote. In the two sharding-based systems RepShard [49] and RepChain [50], reputation is integrated to explicitly characterize heterogeneity among validators, which boots the system throughput by helping elect a high capability leader, while enhancing the system's security via contributing to balancing multiple shards.

To achieve public participation infrastructure maintenance, we have to consider the necessity of real-time interaction and the complexity of involving the public users. The consensus protocol should be low-latency and cost efficient. Besides, it should be compatible with a dynamic public group of verifiers in the blockchain. As shown in Table I, the existing consensus protocols are not applicable in this scenario.

III. System Model

As blockchain-based infrastructure maintenance turns into a promising paradigm for better management and coordination in smart city [51], we consider that a consortium blockchain fits more scenarios, which is in line with several existing research on smart city. Note that the underlying reasons for employing the consortium blockchain are two folds. First, since the population and the number of infrastructures keep increasing, the blockchain for infrastructure maintenance needs to be efficient and scalable, impeding the public blockchain. Second, multiple organizations will be involved in collaboratively implementing the infrastructure maintenance process, leaving the private blockchain inappropriate.

However, directly applying the consortium blockchain to smart-city infrastructure maintenance may not be optimal. As the primary community stakeholders, smart citizens always desire to quickly acquire the latest maintenance updates via witnessing essential procedures, which results in significant challenges for using the traditional consortium blockchain.

To address this concern, we propose a public participation consortium blockchain system as shown in Fig. 1, where a group of public citizens is dynamically selected to participate in the transaction verification process. Thus, the public witness is involved in block generation, which embodies all infrastructure maintenance information.

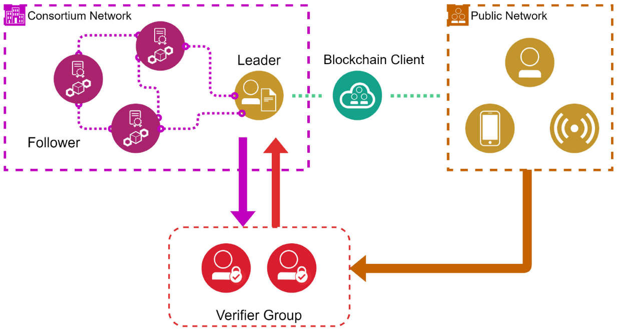


Fig. 1. System overview. The system mainly consists of two parts, consortium cluster and public network. A group of public citizens is dynamically selected to participate in the transaction verification process. Fig. 2 shows the detailed working process of the system.

As illustrated in Fig. 1, the left-most block is the consortium network operated with consortium members, denoted as C, which consists of multiple organizations relevant to infrastructure maintenance, such as the government sectors, municipal companies, etc. We assume that members in the consortium network are trustful. From the perspective of the blockchain operation, one of the consortium members will act as the leader l, and the rest of them are followers $F \triangleq C \setminus \{l\}$. The right-most block indicates the public network, denoted as P, consisting of citizens, RSUs, intelligent terminals, etc. The public users are connected to the consortium nodes through the blockchain client. Moreover, the bottom block is the dynamic verifier group V, which is randomly selected from the public. Technically, verifiers in this group can be regarded as temporary followers in the consortium network via participating in the transaction verification process.

To guarantee the functionality of the blockchain system, a consensus protocol, denoted as the *Protocol*, needs to be designed, which will be detailed in Section IV. Besides, the shared blockchain, denoted as the *Ledger*, is mainly managed by the consortium network, containing the transactions that record the reports of infrastructure damage submitted by users in public network P, and the responses from the authorities. Thus, the "client" submitting the transactions mainly consists of two parts: 1) the public and 2) the authorities. The public part may include citizens, roadside units, intelligent terminals, etc., and the corresponding transactions are related to the

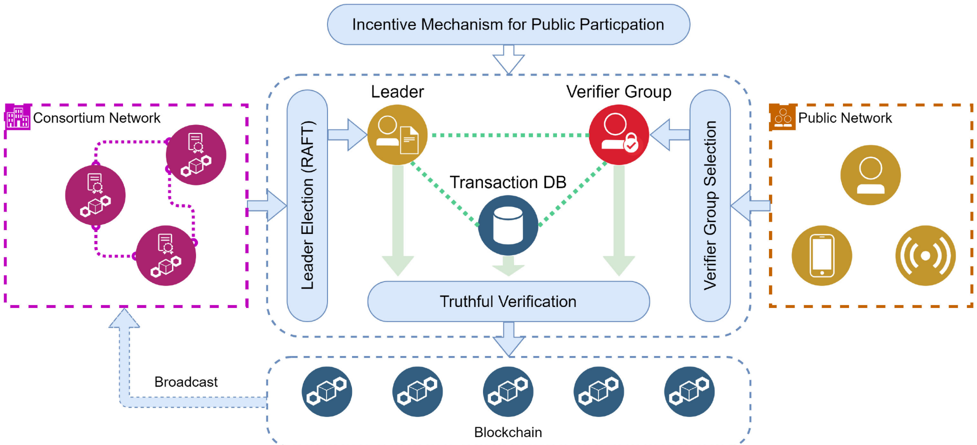


Fig. 2. Working process of the public participation blockchain system for infrastructure maintenance. The whole system mainly has four algorithms, leader election, verifier group selection, truthful verification, and incentive mechanism for public participation.

reports of infrastructure damage. While the authority part may comprise government sectors, municipal companies, etc., and their submitted transactions are the responses of infrastructure damage reports.

We give a formal definition of our proposed public participation blockchain system for smart-city infrastructure maintenance.

Definition 1 (Public Participation Blockchain): The public participation blockchain system is a tuple $<l, F, P, V, Protocol, Ledger>$.

With the inclusion of public participation for transaction verification in blockchain, new challenges appear in our proposed system.

1) *Collusion Attack From Public Verifiers:* Though the public witness brings more democracy and transparency, it brings risks of malicious attacks since the public stakeholders are coming with heterogeneous backgrounds and intentions. Here, the main threat would be a collusion attack where multiple verifiers (in the majority) collude to deliberately accept illegal transactions or reject qualified ones for interest's sake.
2) *Incentive for Public Participation:* Considering that verifying transactions costs both computational and communication resources, public citizens may not actively and continuously join the dynamic verifier group without enough incentive.

To deal with the first challenge, we take advantage of the *private-prior peer-prediction theory* [52], [53], to design a truthful verification mechanism in Section IV-C, where the trustworthiness of each verifier will be evaluated by the leader. To elicit continuous public participation in transaction verification mentioned as the second challenge in the above, we propose to devise an incentive mechanism based on a two-stage Stackelberg game [54]–[56], which will be elaborated in Section IV-D.

IV. Public Participation Blockchain System

In this section, we will present the working process in our proposed public participation blockchain system, including the detailed design of the consensus protocol and an additional incentive mechanism for the involvement of public participation. To denote the state of the blockchain system, we give a definition of *view* as follows.

Definition 2 (View of Blockchain): A view is a tuple of configuration information that identifies a cycle, view $= <l, V, F>$.

The overall process of our proposed public participation blockchain system is presented in Algorithm 1. The blockchain system works in a succession of *views* numbered with monotonically increasing integers. In each view, a unique dedicated *leader* will be elected from consortium members using the Leader Election (line 2). Once the leader is elected, the remaining consortium members will act as *followers*. The leader will lead the consensus process until the view has changed. As the public citizens are designed to participate in the transaction verification process in our system, we consider selecting a *Verifier Group* V with size $|V|$, from the available user set P, which thus can achieve the goal of involving public witness for block generation in an efficient way for reaching consensus. In detail, the leader first gets the list of currently active users from the follower (line 3), and then by executing the Verifier Group Selection function, $|V|$ validators are selected from this *Verifier Group* (line 4). After that, the leader will send *TxList* to the verifier group and followers in the consortium, where the Truthful Verification procedure will be called to collect trustworthy verification and voting results from the verifier group V and the Transaction Verification will be implemented for other official followers, resulting in the jointly verified transactions to generate a new block (line 5). Finally, the leader sends the newly generated block to every followers and thus successfully append it to the main chain (line 6). The whole process is illustrated in Fig. 2.

Detailed designs of the aforementioned subprocedures Leader Election, Verifier Group Selection, and Truthful Verification will be elaborated on in Sections IV-A–IV-C, respectively. To ensure the liveliness of our proposed system, we introduce an incentive mechanism for public participation in Section IV-D.

Algorithm 1 Overall Process of Our Proposed Blockchain System

```
1: while ViewNumber $T$ do
2:     leader, follower ← LEADER ELECTION(T)
3:     $P$ ← leader requests followers to report active public users
4:     Verifier Group ← VERIFIER GROUP SELECTION($P$)
5:     NewBlock ← TRUTHFUL VERIFICATION(TxList, Verifier Group) and TRANSACTION VERIFICATION(TxList, followers)
6:     leader broadcasts NewBlock to the consortium chain
7: end while
```

A. Leader Election

We adopt the proven Raft leader election algorithm [44] to elect the leader of the consortium blockchain. Raft uses randomized timers to elect leaders. As shown in Algorithm 2, members in the consortium blockchain network have three roles: 1) *leader*; 2) *candidate*; and 3) *follower*. Raft uses a heartbeats mechanism to trigger leader election. When starting up, they begin as followers (lines 8–14). A node continues to be a follower as long as it receives valid requests from a leader or candidate. The leader sends periodic heartbeats to all followers in order to maintain its authority (lines 1–7). If a follower receives no communication over a period of time *node.timeout*, then it assumes that there is no viable leader and begins an election to choose a new leader. To begin an election, a follower increases its current term *node.term* and transits to the candidate state (lines 16–30). It then votes for itself and issues *RequestVote* in parallel to others. A candidate continues in this state until one of the following three cases happens.

1) It wins the election. A candidate wins an election if it receives votes from a majority of the nodes for the same term.
2) Another node claims to be the leader. While waiting for votes, a candidate may receive a message from another node being the leader. If this leader's term is at least as large as the candidate's current term, then the candidate recognizes the leader to be legitimate.
3) A period of time goes by with no winner.

B. Verifier Group Selection

In this section, we depict how the *Verifier Group* is determined. As shown in Algorithm 3, the leader first decides a list of verifier candidates based on the following criteria (line 1).

1) The candidates are selected based on their location information. The leader figures out the location information contained in the current *TxList* and then choose citizens near these locations as verifier candidates.
2) When the location information contained in the current *TxList* is scattered in several distant areas, it is necessary to ensure that the number of candidates related to any specific area is not less than 20% of the total number of candidates for verification effectiveness consideration.

Algorithm 2 Leader Election

```
Input: ViewNumber
Output: leader, follower
1: if node.role = leader then
2:     broadcast an initial heartbeat
3:     if receive message contains view number $T$ > node.ViewNumber then
4:         node.ViewNumber ← $T$
5:         node.role ← Follower
6:     end if
7: end if
8: if node.role = follower then
9:     if receive a message from leader or candidate then
10:        reset node.timeout
11:    end if
12:    if node.timeout countdown to zero then
13:        node.role ← Candidate
14:    end if
15: end if
16: if node.role = candidate then
17:    do following procedure ELECTION
18:    procedure ELECTION
19:        increase node.ViewNumber
20:        vote for self
21:        reset node.timeout
22:        broadcast message RequestVote
23:    end procedure
24:    wait for votes from other nodes:
25:        Case 1 received votes > $\frac{2}{3}|V|$
26:            node.role ← leader
27:        Case 2 receive message from the new leader
28:            node.role ← follower
29:        Case 3 node.timeout countdown to zero
30:            start new ELECTION
31: end if
```

Algorithm 3 Verifier Group Selection

```
Input: public user set $P$
Output: Verifier Group
1: CandidatesList ← leader select from $P$
2: leader broadcast CandidatesList
3: leader request follower vote for CandidatesList
4: if received votes > $\frac{2}{3}|F|$ then
5:     Verifier Group ← leader modify candidateslist
6: end if
7: return Verifier Group
```

When a sufficient number of candidates are selected, the *CandidatesList* is shared with all followers in the consortium network (line 2). Followers are asked to vote for this verifier list and check whether the candidates are still active (line 3). If there are 2/3 followers agreed on the CandidatesList, the leader can determine verifiers on this list forming the *Verifier Group* and the inactive candidates will be kicked out (line 5).

C. Truthful Verification Based on Peer Prediction

As mentioned in Section III, the public witness brings risks of malicious attacks. To deal with this challenge, the private-prior peer prediction theory is utilized to build the truthful verification mechanism. Private-prior peer prediction is an incentive-compatible mechanism for eliciting truthful reports. Different from the classical peer prediction method, this approach relaxes the requirement of common prior amongst both the participants and the mechanism, in which each participant could keep a subjective and private belief. In detail, each participant coupled with a reference peer is required to submit their prior and posterior beliefs. According to these two reports, the trustworthiness of each participant could be calculated using a strictly proper scoring rule, which can be used to detect malicious behavior and encourage participants to report honestly.

Denote the size of the dynamic verifier group as N. The leader will broadcast unverified transactions to all the public verifiers. We consider that each transaction has an inherent characteristic, denoted by Q, which is regarded as a random variable represented by l, h. For example, this characteristic could tell whether the transaction is fake or not, in this case, $Q = l$ indicates the transaction is fake, and $Q = h$ means the transaction is authorized. Also, this characteristic could show whether the transaction is related to a specific group's interest, in this case, $Q = h$ indicates that the transaction is important to this group.

When the transaction has been verified, verifier i generates a binary opinion denoted by $S_i = s_i \in \{l, h\}$. S_i is verifier $i's$ personal judgment on the characteristic Q of the transaction. In the aforementioned first case, $S_i = l$ indicates that verifier i thinks this transaction is invalid. The *opinion report* of verifier i, denoted by $x_i \in \{0, 1\}$, is generated by applying a report strategy $r_i : S_i \longrightarrow \{0, 1\}$. For honest verifier i, she will report $x_i = 1$ when $S_i = h$, and $x_i = 0$ otherwise.

1) Prior Belief Reports: To calculate the trustworthiness of verifier i, the leader will randomly choose a reference verifier $j \neq i$. Before starting to verify the transaction, verifier i is required to report her prior belief $y_{ij} \in [0, 1]$ to the leader that her reference peer j will report a high signal, i.e., $x_j = 1$.

2) Posterior Belief Reports: After verifying the transaction m, verifier i makes her own judgment on this transaction $S_i = s_i$. Then, verifier i needs to send her posterior belief, denoted by $y'_{ij} \in [0, 1]$, that her reference peer j will report personal judgment on transaction as $x_j = 1$.

Verifier $i's$ posterior belief y'_{ij} can be expressed as

$$\begin{aligned} y'_{ij}(s_i) &= P_i\big(x_j = 1 | S_i = s_i\big) \\ &= P\big(x_j = 1 | Q = h\big) P(Q = h | S_i = s_i) \\ &\quad + P\big(x_j = 1 | Q = l\big) P(Q = l | S_i = s_i). \end{aligned}$$

3) Verifier's Trustworthiness: Based on reports y_{ij} and $y'_{ij}(s_i)$, the leader calculates verifier i's trustworthiness through a certain scoring rule. Verifiers with low trustworthiness levels are classified as malicious, and their reports will not be considered anymore. Here, we use a *strictly proper scoring rule* [52], which can motivate users to provide truthful reports y_{ij}, and $y'_{ij}(s_i)$. Specifically, we give the following definition.

Definition 3 (Strictly Proper Scoring Rule): A binary scoring rule is proper if it leads to agent maximizing her score by truthfully providing her report $y \in [0, 1]$, and is strictly proper if an agent can maximize her score if and only if providing her report truthfully.

Taking the binary logarithmic scoring rule as an example, we have

$$R(y, \omega = 1) = \ln y \tag{1a}$$

$$R(y, \omega = 0) = \ln (1 - y) \tag{1b}$$

where ω indicates the binary opinion report.

Before we give the expression of verifier $i's$ trustworthiness, it is notable that in basic private prior peer prediction, verifier i is required to send reports y_{ij}, $y'_{ij}(s_i)$, but not her opinion report x_i. Instead of using verifiers' original opinion report, the leader infers opinion report x_i according to verifier $i's$ reports y_{ij} and $y'_{ij}(s_i)$.

Inferred opinion report x_i is generated by applying

$$x_i = x\Big(y_{ij}, y'_{ij}\Big) = \begin{cases} 1, & y_{ij} < y'_{ij} \\ 0, & y_{ij} > y'_{ij}. \end{cases} \tag{2}$$

Then, we can define the trustworthiness of verifier i as a function of y_{ij}, $y'_{ij}(s_i)$, and x_j

$$T_i = \alpha R\big(y_{ij}, x_j\big) + (1 - \alpha) R\Big(y'_{ij}, x_j\Big) + \beta \tag{3}$$

where $R(y, \omega)$ is a strictly proper scoring rule mentioned above, and $\alpha \in [0, 1]$ is a parameter weighing the importance of the prior and posterior belief.

To keep the budget balance, β could be given by

$$\beta = -\frac{1}{N} \sum_{k=1}^{N} \Big[\alpha R\big(y_{kj}, x_j\big) + (1 - \alpha) R\Big(y'_{kj}, x_j\Big) \Big].$$

Note that the trustworthiness of verifier i is determined on verifier j's inferred opinion report x_j, verifier i's prior belief y_{ij} report, and posterior belief report $y'_{ij}(s_i)$. It means that one verifier's trustworthiness is irrelevant to reports of the other verifiers in the system. Therefore, the cooperative cheating of malicious verifiers will have little effect on the evaluation of trustworthiness.

4) Transaction Score: After receiving every verifiers' reports, the leader calculates verifiers' trustworthiness according to (1a), (1b), (2), and (3).

Then, the leader can calculate score for transaction m

$$\text{Score}(m) = \sum_{i \in B} x_i * T_i \tag{4}$$

where $B = \{i | T_i > t\}$ is the set of honest verifiers with high trustworthiness T_i; the system parameter t is set as the threshold of trustworthiness.

Besides, according to the score, the leader will decide whether a transaction is permitted to be included in a new block. A simple decision-making rule is that a transaction will be included if its score is larger than $(|B|/3)$.

Algorithm 4 Truthful Verification

```
Input: TxList, Verifier Group
Output: NewBlock
leader send TxList to Verifier Group
$\{y, y'\} \leftarrow$ leader ask each verifier to report her prior belief and posterior belief
for all verifier $i \in$ Verifier Group do
    i.peer $\leftarrow$ leader specify a reference peer for verifier $i$
    i.opinion $\leftarrow$ leader implicit verifier $i$'s opinion report
    i.score $\leftarrow$ leader calculate score for verifier $i$
end for
for all $tx \in$ TxList do
    tx.score $\leftarrow$ leader calculate priority for $tx$
end for
Block $\leftarrow$ leader packs transaction with high score
leader request follower vote for Block
if received votes $> \frac{2}{3}|F|$ then
    NewBlock $\leftarrow$ Block
end if
return NewBlock
```

5) Peer-Prediction-Based Truthful Verification Process: In summary, we can describe the overall process of the proposed truthful transaction verification scheme as follows, which is also presented with the pseudocode in Algorithm 4.

1) For every verifier i, the leader randomly chooses another nonoverlapped verifier j as her reference peer. Then, a list of transactions is sent to verifier i.
2) Verifier i is asked to report her prior belief $y_{ij} \in [0, 1]$ the probability her reference peer j will provide a report to the leader that j evaluates the transaction m with high priority.
3) Verifier i verifies each transaction and then makes her judgment $S_i = s_i$ for each transaction.
4) Verifier i is asked to report for her posterior belief report $y'_{ij} \in [0, 1]$, with $y_{ij} \neq y'_{ij}$.
5) The leader infers the opinion report x_i, x_j of verifier i and verifier j, respectively, then the leader calculates verifiers $i's$ trustworthiness according to (3).
6) The leader calculates the score for each transaction based on (4) and sorts all the transactions according to their score.
7) The leader packs transactions, whose score is larger than $(|B|/3)$ into a new block. In other words, a transaction is permitted to be included in a block if and only if more than half of honest verifiers agree.
8) The leader requests followers in the consortium vote for the new block. A new block is permitted to be appended to the main chain if and only if over (2/3) followers agree.

6) Proof of Incentive Compatibility:

Theorem 1: The proposed peer-prediction-based truthful verification scheme is incentive compatible, where the truthful report $y_{ij} = Pr_i(x_j = h)$ and $y'_{ij} = Pr_i(x_j = h|S_i = s_i)$ is the optimal strategy.

Proof: Since we require that each verifier should report their prior belief y_{ij} and posterior y'_{ij} before and after making judgment $S_i = s_i$, y_{ij} and y'_{ij} are independent and then we have

$$\begin{aligned} E[T_i] &= E\big[\alpha R\big(y_{ij}, x_j\big)\big] + E\Big[(1-\alpha)R\Big(y'_{ij}, x_j\Big)\Big] \\ &= \alpha E\big[R\big(y_{ij}, x_j\big)\big] + (1-\alpha)E\Big[R\Big(y'_{ij}, x_j\Big)|S_i = s_i\Big]. \end{aligned}$$

We employ the binary logarithmic scoring rule.

Let $p_1 = Pr_i(x_j = h)$ and $p_2 = Pr_i(x_j = h|S_i = s_i)$, and then we have

$$\begin{aligned} E[T_i] = {} & \alpha\big(p_1 \ln y_{ij} + (1-p_1)\ln\big(1-y_{ij}\big)\big) \\ & + (1-\alpha)\Big(p_2 \ln y'_{ij} + (1-p_2)\ln\Big(1-y'_{ij}\Big)\Big). \end{aligned}$$

For any i and j, take the partial derivatives with respect to y_{ij} and y'_{ij}

$$\begin{aligned} \frac{\partial E[T_i]}{\partial y_{ij}} &= \alpha \frac{p_1 - y_{ij}}{y_{ij}\big(1-y_{ij}\big)}, \\ \frac{\partial E[T_i]}{\partial y'_{ij}} &= (1-\alpha)\frac{p_2 - y'_{ij}}{y'_{ij}\Big(1-y'_{ij}\Big)}. \end{aligned}$$

Therefore, we get the zero points as

$$\begin{aligned} \hat{y_{ij}} &= p_1 = Pr\big(x_j = h\big) \\ \hat{y'_{ij}} &= p_2 = Pr\big(x_j = h|S_i = s_i\big). \end{aligned}$$

Then, take the second partial derivatives with respect to y_{ij} and y'_{ij}

$$\begin{aligned} \frac{\partial^2 E[T_i]}{\partial y_{ij}^2} &= \alpha \frac{-y_{ij}^2 + 2p_1 y_{ij} - p_1}{y_{ij}^2\big(1-y_{ij}\big)^2}, \\ \frac{\partial^2 E[T_i]}{\partial y_{ij}'^2} &= (1-\alpha)\frac{-y_{ij}'^2 + 2p_2 y'_{ij} - p_2}{y_{ij}'^2\Big(1-y'_{ij}\Big)^2} \end{aligned}$$

let $y_{ij} = \hat{y_{ij}}$ and $y'_{ij} = \hat{y'_{ij}}$, then we have

$$\begin{aligned} \left.\frac{\partial^2 E[T_i]}{\partial y_{ij}^2}\right|_{y_{ij}=\hat{y_{ij}}} &< 0 \\ \left.\frac{\partial^2 E[T_i]}{\partial y_{ij}'^2}\right|_{y'_{ij}=\hat{y'_{ij}}} &< 0. \end{aligned}$$

Therefore, the maximum of $E[y_{ij}]$ can be achieved when $y_{ij} = Pr_i(x_j = h)$ and $y'_{ij} = Pr_i(x_j = h|S_i = s_i)$, which means that user i can receive the maximum trustworthiness if and only if she reports both y_{ij} and y'_{ij} honestly. *Binary quadratic scoring rule:* We employ the binary quadratic scoring rule. Let $p_1 = Pr_i(x_j = h)$ and $p_2 = Pr_i(x_j = h|S_i = s_i)$, thus we have

$$\begin{aligned} E[T_i] = {} & \alpha\Big(p_1\Big(2y_{ij} - y_{ij}^2\Big) + (1-p_1)\Big(1 - y_{ij}^2\Big)\Big) \\ & + (1-\alpha)\Big(p_2\Big(2y'_{ij} - y_{ij}'^2\Big) + (1-p_2)\Big(1 - y_{ij}'^2\Big)\Big). \end{aligned}$$

Taking the partial derivatives with respect to y_{ij} and y'_{ij}, we could get the zero point $y_{ij} = Pr_i(x_j = h)$ and $y'_{ij} = Pr_i(x_j = h|S_i = s_i)$. Since the second partial derivatives

$$\frac{\partial^2 E[T_i]}{\partial y_{ij}^2} = -2\alpha$$
$$\frac{\partial^2 E[T_i]}{\partial y_{ij}'^2} = -2(1-\alpha)$$

are always negative, the maximum of $E[y_{ij}]$ could be achieved when $y_{ij} = Pr_i(x_j = h)$ and $y'_{ij} = Pr_i(x_j = h|S_i = s_i)$. ■

D. Incentive Mechanism for Public Participation

In this section, we proposed an incentive mechanism to encourage the public to participate in the truthful verification. We formulate the actions of public participants and the blockchain as a two-stage Stackelberg game, where their equilibrium strategies are derived to achieve incentive goals.

1) Utility Function: The leader in blockchain aims to encourage citizens in the verifier Group to verify transactions and report their prior beliefs. Here, we assume that the leader distributes a total payment $\tau > 0$ for the public verifiers verifying transactions in each block. According to the leader's payment announcement, each selfish and rational verifier chooses the effort level of their participation, such as the number of transactions verified, which will affect the utilities of both the leader and the verifier.

For simplicity, let $z_i \in [0, D]$ denote the number of transactions that verifier i has verified, where D is the number of transactions sent to the verifier. Let C_i be the nonnegative unit computation and communication cost, then the reward received by verifier i is proportional to z_i, which can be defined as

$$u_i(z_i, \mathbf{z}_{-i}) = \frac{z_i}{\sum_{k=1}^{N} z_k}\tau - z_i C_i. \tag{5}$$

In the above equation, $\mathbf{z}_{-i} = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_N)$ denotes the strategies of other verifiers except for verifier i and $N = |V|$ is the size of the verifier Group.

Let $u(\tau)$ denote the utility of the leader, which consists of the gain due to verifiers' participation and the total payment cost to verifiers. Intuitively, the gain received by the leader is proportional to verifiers' actions $\mathbf{z} = (z_1, \ldots, z_N)$, which is denoted as $g(\mathbf{z})$. Therefore, the utility of the leader can be defined as

$$u(\tau) = \lambda g(\mathbf{z}) - \tau \tag{6}$$

where $\lambda > 0$ is a system parameter. We take the *Sigmoid* function as an example to model $g(\mathbf{z})$ as follows:

$$g(\mathbf{z}) = \frac{1}{1 + e^{-\sum_{i=1}^{N} z_i}}.$$

Generally, the larger the total number of verified transactions from the verifier group, the higher the gain of the leader; while when the number of verified transactions is too small or too large, the gain of the leader will not increase that quickly.

In summary, we formulate the incentive mechanism for public participation as a two-stage Stackelberg game in each term. In the first stage, the leader decides to maximize $u(\tau)$, and in the second stage, each verifier chooses the participation level according to the Stackelberg equilibrium strategy to maximize its utility.

2) Equilibrium Analysis:

Definition 4 (Stackelberg Equilibrium Strategy): Stackelberg equilibrium strategies denoted by ($\mathbf{z}^*$, τ^*) consist of the best response of each player in the Stackelberg game, where

$$z_i^* = \arg \max_{z_i \in [0,D]} u_i(z_i, \mathbf{z}_{-i})$$
$$\tau^* = \arg \max_{\tau > 0} u(\tau).$$

Note that the second stage of the game can be considered as a noncooperative game. For any payment strategy τ given by the leader and other verifiers strategies $\mathbf{z}_{-i}$, verifier i would like to determine an optimal strategy z_i to maximize its own utility.

Definition 5 (Nash Equilibrium): A set of strategies $\mathbf{z}^* = (z_1^*, \ldots, z_N^*)$ is a Nash equilibrium of the second stage if for each verifier i and any z_i, there exists

$$u_i\big(z_i^*, \mathbf{z}_{-i}^*\big) \geq u_i\big(z_i, \mathbf{z}_{-i}^*\big).$$

Theorem 2: For each verifier i participating in the verification process, the optimal strategy is

$$z_i^* = \frac{(N-1)\tau}{\sum_{j=1}^{N} C_j}\left(1 - \frac{(N-1)C_i}{\sum_{j=1}^{N} C_j}\right).$$

Proof: To study the Nash equilibrium of the second stage of the game, we derive the first-order derivative of $u_i(z_i, \mathbf{z}_{-i})$ with respect to z_i as

$$\frac{\partial u_i(z_i, \mathbf{z}_{-i})}{\partial z_i} = \frac{\tau}{\sum_{j=1}^{N} z_j} - \frac{\tau z_i}{\left(\sum_{j=1}^{N} z_j\right)^2} - C_i$$

based on which we can derive the second-order derivative of $u_i(z_i, \mathbf{z}_{-i})$ with respect to z_i as

$$\frac{\partial^2 u_i(z_i, \mathbf{z}_{-i})}{\partial^2 z_i} = -\frac{\tau \sum_{j=1, j\neq i}^{N} z_j}{\left(\sum_{j=1}^{N} z_j\right)^3} < 0.$$

Therefore, the utility of verifier i $u_i(z_i, \mathbf{z}_{-i})$ is a strictly concave function for $z_i \in [0, D]$. That is, there exists z_i^* that maximizes u_i.

Setting $[(\partial u_i(z_i, \mathbf{z}_{-i}))/(\partial z_i)] = 0$, we can obtain

$$z_i^* = \sqrt{\frac{\tau \sum_{j=1, j\neq i}^{N} z_j^*}{C_i}} - \sum_{j=1, j\neq i}^{N} z_j^*. \tag{7}$$

If the right-hand side of (7) is positive, it is the optimal strategy of verifier i due to the concavity of u_i. Otherwise, verifier i does not participate in the verifying process. Furthermore, if the right-hand side of (7) is more than D, the optimal strategy of verifier i is $z_i = D$.

According to (7), for any verifier i, we have

$$\sum_{j=1}^{N} z_j^* = \sqrt{\frac{\tau \sum_{j=1, j\neq i}^{N} z_j^*}{C_i}}. \tag{8}$$

By setting $\hat{z} = \sum_{j=1}^{N} z_j^*$ and squaring both sides of (8), we can derive that

$$\begin{cases} z_1^* = \hat{z} - \frac{\hat{z}^2 C_1}{\tau} \\ \vdots \\ z_i^* = \hat{z} - \frac{\hat{z}^2 C_i}{\tau} \\ \vdots \\ z_N^* = \hat{z} - \frac{\hat{z}^2 C_N}{\tau}. \end{cases} \tag{9}$$

Adding up the above equations, we have

$$\hat{z} = N\hat{z} - \frac{\hat{z}^2 \sum_{j=1}^{N} C_j}{\tau}. \tag{10}$$

Solving $\hat{z}$ in (10), we obtain

$$\hat{z} = \frac{\tau(N-1)}{\sum_{j=1}^{N} C_j}. \tag{11}$$

By plugging (11) into (9), we can derive

$$z_i^* = \frac{(N-1)\tau}{\sum_{j=1}^{N} C_j}\left(1 - \frac{(N-1)C_i}{\sum_{j=1}^{N} C_j}\right).$$

■

According to the above analysis, the leader knows that there exists a unique Nash equilibrium among verifiers under any given τ. Therefore, the leader could maximize the utility by choosing the optimal τ. We have

$$u(\tau) = \frac{\lambda}{1 + e^{-\sum_{i=1}^{N} z_i}} - \tau. \tag{12}$$

Theorem 3: There exists a unique τ^* maximizes the utility of the leader for $\tau \in [0, \infty)$.

Proof: The first-order derivative of $u(\tau)$ with respect to τ is

$$\begin{aligned} \frac{\partial u(\tau)}{\partial \tau} &= \lambda g'(\mathbf{z}^*)\frac{\partial \mathbf{z}^*}{\partial \tau} - 1 \\ &= \lambda g'(\mathbf{z}^*)\left(\frac{\partial z_1^*}{\partial \tau} + \cdots + \frac{\partial z_N^*}{\partial \tau}\right) - 1 \\ &= \frac{1-N}{\sum_{j=1}^{N} C_j} \frac{\lambda e^{-\sum_{i=1}^{N} z_i^*}}{\left(1 + e^{-\sum_{i=1}^{N} z_i^*}\right)^2} - 1. \end{aligned}$$

Hence, the second-order derivative of $u(\tau)$ with respect to τ is

$$\begin{aligned} \frac{\partial^2 u(\tau)}{\partial^2 \tau} &= \lambda g'(\mathbf{z}^*)\frac{\partial^2 \mathbf{z}^*}{\partial \tau^2} + \lambda g''(\mathbf{z}^*)\left(\frac{\partial \mathbf{z}^*}{\partial \tau}\right)^2 \\ &= \lambda\left(\frac{N-1}{\sum_{j=1}^{N} C_j}\right)^2 \frac{-e^{-\sum_{i=1}^{N} z_i^*}}{\left(1 + e^{-\sum_{i=1}^{N} z_i^*}\right)^2} \\ &\quad + \lambda\left(\frac{N-1}{\sum_{j=1}^{N} C_j}\right)^2 \frac{2e^{-2\sum_{i=1}^{N} z_i^*}}{\left(1 + e^{-\sum_{i=1}^{N} z_i^*}\right)^3} \\ &< 0. \end{aligned}$$

Therefore, the utility of leader $u(\tau)$ defined in (12) is a strictly concave function for $\tau \in [0, \infty)$. Let K denote $[(\lambda(N-1))/(\sum_{j=1}^{N} C_j)]$.

For the case $K < 4$, the first-order derivative of $u(\tau)$ always be negative. Therefore, in this case, the optimal strategy for leader is $\tau^* = 0$.

For the case $K > 4$, setting $[(\partial u(\tau))/(\partial \tau)] = 0$, we obtain

$$\tau^* = -\frac{\lambda}{K} \ln \frac{K - 2 - \sqrt{K(K-4)}}{2}.$$

■

Finally, we get the optimal strategies $(\mathbf{z}^*, \tau^*)$ for each verifier and leader, respectively

$$z_i^* = \begin{cases} 0, & \tau \le C_i \sum_{j=1, j \neq i}^{N} z_j^* \\ \frac{(N-1)\tau}{\sum C_j}\left(1 - \frac{(N-1)C_i}{\sum C_j}\right), & z_i^* \in (0, D) \\ D, & \text{otherwise} \end{cases}$$

$$\tau^* = \begin{cases} 0, & \frac{\lambda(N-1)}{\sum C_j} < 4 \\ -\frac{\lambda}{K} \ln \frac{K-2-\sqrt{K(K-4)}}{2}, & \frac{\lambda(N-1)}{\sum C_j} > 4. \end{cases}$$

V. Experimental Evaluation

In this part, we perform simulation experiments to analyze the properties and performances of our proposed public participation consortium blockchain system. The performance of the whole system is impacted by the three main proposed algorithms, such as the peer-prediction-based trustworthiness value system, the truthful-verification-based consensus scheme, and the Stackelberg-game-based incentive mechanism. Thus, we perform simulation experiments to analyze the properties and performances of these three main schemes and the holistic consensus protocol instead of the consortium blockchain system. Basically, in order to simulate possible actions of users in real life, we divided the dynamic verifier group into three types: 1) reliable honest verifiers; 2) unreliable honest verifiers; and 3) malicious ones. To evaluate the trustworthiness value system, we calculated the accumulated trustworthiness values of different kinds of verifiers and showed that the trustworthiness value can reliably distinguish honest verifiers from dishonest verifiers. Then, to evaluate the incentive mechanism, we studied the influence of several system parameters, such as the verifier group size N, on the optimal strategies and rewards of the leader and the verifier. Finally, we compared the accuracy and efficiency of our proposed consensus protocol with another two consensus protocols, which are the variants of our scheme. The detailed settings are discussed in each section. All experiments are conducted by using MATLAB R2020b on a Windows10 desktop with a Ryzen 3600X Processor and 16-GB RAM.

A. Experiment Settings

For the whole blockchain system, we set the number of followers as $|F| = 150$ and the size of the verifier group as $N = |V| = 100$ unless otherwise specified. To simulate various features of public participants, we consider the following three types of verifiers in our system.

1) *Reliable Honest Verifiers:* They have high judgment accuracy rates and will send honest reports about the received transactions, which are the most preferred verifiers.

2) *Malicious Verifiers:* They have high judgment accuracy rates but tend to submit reports untruthfully.
3) *Unreliable Honest Verifiers:* They have low judgment accuracy rates but always tell the truth.

Note that here we consider the consortium members in the blockchain to be reliable honest. These three types of verifiers exist in the public network with a certain proportion. We set that the probability of misjudgment P_{mj} is a uniform distribution variable, where reliable honest and malicious verifiers have $P_{mj} \sim U[0.1, 0.2]$ and unreliable honest verifiers have $P_{mj} \sim U[0.3, 0.4]$. Besides, we set the probability of misreporting for malicious verifiers $P_{mr} = 0.45$. We assume that all honest verifiers always report truthfully, i.e., $P_{mr} = 0$.

To evaluate the accuracy and efficiency of our proposed consensus protocol, we compare it with other two consensus protocols, i.e., traditional consortium consensus and joint consensus, which are introduced as follows.

1) *Consortium Consensus:* Consortium members consisting of the leader and the rest of the followers reach a consensus on all events that happened on the blockchain. In particular, transactions are permitted to be included in a block and then appended to the main chain if and only if the majority of the followers agree.
2) *Joint Consensus:* Compared to the consortium consensus, a public verifier group is involved in reaching a consensus on transaction validation. A transaction is permitted to be included in a block if and only if the majority of the followers in the consortium and the majority of the verifier group members agree. After that, a new block is permitted to be appended to the main chain if and only if the majority of the followers agree.

B. Evaluation of Peer-Prediction-Based Truthful Verification

We consider that the trustworthiness of each verifier is accumulated as time goes on. To calculate the trustworthiness in different forms, two scoring rules, i.e., binary logarithmic and binary quadratic, are studied here. Simulation results of verifiers' accumulated trustworthiness in 200 rounds of truthful verification processes are shown in Fig. 3, in which we assume that the percentages of the reliable honest verifier, unreliable honest verifier, and malicious verifier in the overall verifier group are 40%, 40%, and 20%, respectively, and the results of two randomly selected verifiers from each type are presented. Fig. 3 indicates that the accumulated trustworthiness of honest verifiers increases with the rounds, in the long term. On the contrary, the accumulated trustworthiness of malicious verifiers is on a downward trend and is generally negative. Visually, the curves of malicious verifiers have a greater inclination than honest verifiers; it reveals that the penalty for malicious behavior is much greater than the reward for honest behavior, which makes the dishonest action more costly. In Fig. 3, we can find that at some points, the accumulated trustworthiness of honest verifiers decreases, while the accumulated trustworthiness of malicious verifiers increases. This indicates that honest verifiers get a negative trustworthiness value at some points, and malicious verifiers even get a positive trustworthiness value. Since the trustworthiness is irrelevant to inferred reports of the selected peer in the system, when, by some coincidence, the peer selected by the system as an honest verifier is a malicious verifier, then the behavior of the honest one will be misjudged as dishonest and be punished. Similarly, when a malicious verifier's peer happens to be malicious, the system will also make a wrong judgment, thus giving the malicious verifier a positive trustworthiness value. Besides, as shown in Fig. 3(a) and (b), no matter which scoring rule is applied, the accumulative trustworthiness shows a similar tendency.

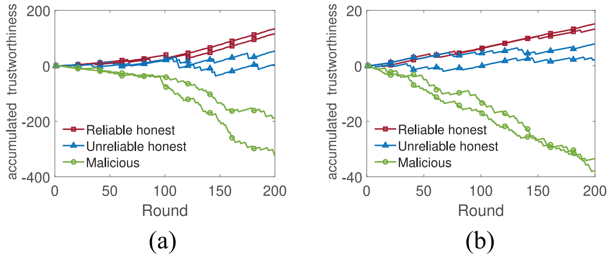


Fig. 3. Accumulative trustworthiness of user sample. (a) Binary logarithmic scoring, $\alpha = 0.5$. (b) Binary quadratic scoring, $\alpha = 0.5$.

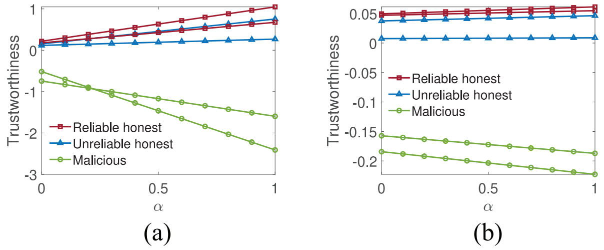
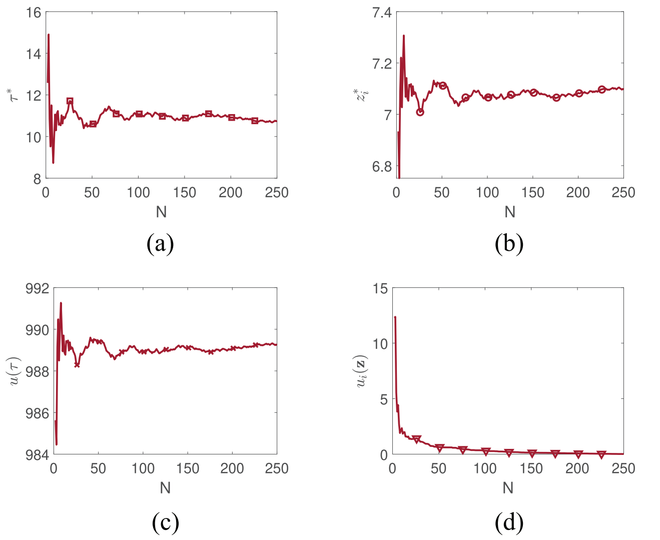


Fig. 4. Influence of α on users' trustworthiness. (a) Binary logarithmic scoring. (b) Binary quadratic scoring.

Next, we evaluate the impact of the system parameters α, weighing the importance of the prior and posterior belief, on the trustworthiness of verifiers. In general, the greater α value, the greater the importance of the posterior belief when calculating verifiers' trustworthiness. Fig. 4(b) and (a) presents how the average trustworthiness changes when it increases using the binary quadratic and logarithmic scoring rules, respectively. As depicted in Fig. 4, the average value of trustworthiness increases with the increase of α for honest verifiers, while the average trustworthiness of the malicious verifier decreases rapidly when the importance of posterior belief becomes larger. It indicates that the verifier's posterior belief is the key factor to detect the malicious verifier. The difference between the verifier's prior report and the posterior report is the key point in observing the psychological changes of the verifier. Since the prior report is a benchmark for measuring this difference, giving more weight to the posterior report could result in a larger gap between the trustworthiness of an honest verifier and a malicious one.

C. Evaluation of the Stackelberg-Based Incentive Mechanism

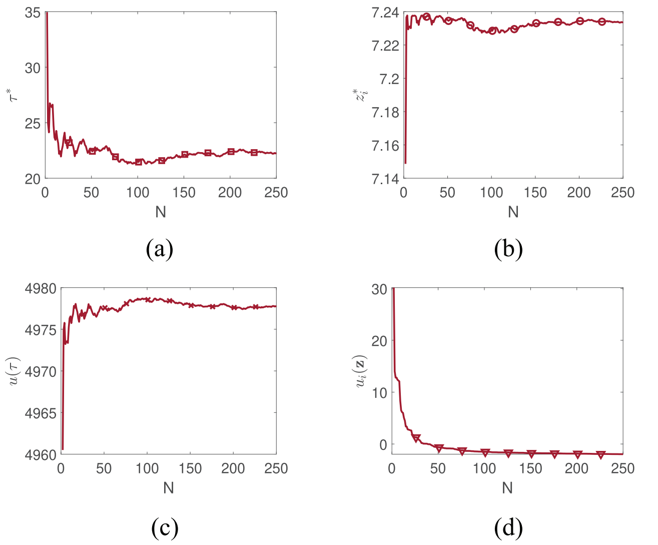
In this section, we evaluate the performance of our proposed Stackelberg-based incentive mechanism for the public verifiers by studying the influence of the verifier group size N on the



(a) (b)

(c) (d)

Fig. 5. Stackelberg equilibrium and utility with $\lambda = 1000$. (a) Leader's optimal strategy. (b) Verifier's optimal strategy. (c) Leader's utility. (d) Verifier's utility.



(a) (b)

(c) (d)

Fig. 6. Stackelberg equilibrium and utility with $\lambda = 5000$. (a) Leader's optimal strategy. (b) Verifier's optimal strategy. (c) Leader's utility. (d) Verifier's utility.

optimal strategies and rewards of the leader and the verifier. Here, we set the nonnegative unit computational cost and communication cost C of each verifier as a uniform distribution variable, i.e., $C \sim U[3, 5]$.

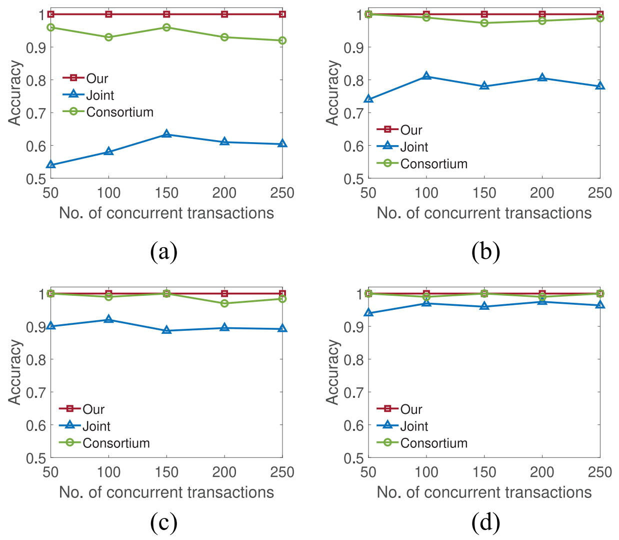
Fig. 5 presents how the optimal strategies and rewards of the leader and verifiers change when the size of the verifier group increases with the system parameter $\lambda = 1000$. When the validator group is relatively small, as depicted in Fig. 5(a) and (b), the leader's optimal strategy τ^* gradually drops while the verifier's optimal strategy z_i^* increases. The underlying reason is that when the size of the verifier group is relatively small, the leader needs to use higher salaries to attract citizens to participate in transaction verification as much as possible to ensure that the work can be completed as expected. For validators, the competition within the verifier group is small, so that even if they reduce their workload, their income can also be guaranteed, as depicted in Fig. 5(d), leading to high income for the verifier. With the growth of the verifier group size, the competition among verifiers is getting stronger so they need to work harder to ensure their income, while the leader does not need to announce huge salaries to attract verifiers anymore. The decrease in total salary and the growth of the verifier group have led to a rapid decline in the income of validators at this stage, as shown in Fig. 5(d). Since the leader's reward is positively correlated with the verifier's participation, Fig. 5(c) shows the same tendency with Fig. 5(b). When the group of validators becomes larger, equilibrium has been reached. To maximize their interests, even if the validator group size is still increasing, the leader and verifier are unwilling to change their strategies, and as a result, curves in Fig. 5(a) and (b) converge to a stable value. However, the curve in Fig. 5(c) tends to be a stable value due to the reason that the work provided by the verifiers has been oversaturated. Unlike the other three figures, the curve in Fig. 5(d) still maintains a downward trend, since the total salary provided by the leader no longer changes and the reward for each validator's work could only decrease as the verifier group increases.

Fig. 6 reports similar results when $\lambda = 5000$. Due to the special position of λ in the leader's utility function IV-D1, it has the most significant impact on the change of the value in Fig. 6(c). It can be seen that the leader's utility has also increased roughly 5 times. Similarly, the values in the other three figures are also increasing. However, the most significant change brought by the increase of λ that the curve in each subfigure gets smoother, and the verifier's strategy reaches a steady-state earlier, which indicates that the increase of λ effectively eliminates the instability in the system resulting from the randomness of the verifier's computation and communication costs.

D. Performance Comparison of Different Consensus Protocols

Finally, we report the accuracy and efficiency of three consensus protocols, i.e., our proposed consensus, consortium consensus, and joint consensus, given different numbers of transactions and followers, where the number of transactions increases from 50 to 250 with an interval of 50 and the number of followers increases from 50 to 200 with the same interval value. To simulate the real situation, artificial transactions are randomly generated at a rate of 20%. Fig. 7 shows the accuracies of different consensus protocols in packing truthful transactions into blocks. The joint consensus protocol shows the worst result, while the consortium consensus protocol achieves relatively high accuracy but still cannot guarantee 100% accuracy. In contrast, our proposed public participation consensus with truthful verification can always record all truthful transactions into blocks.

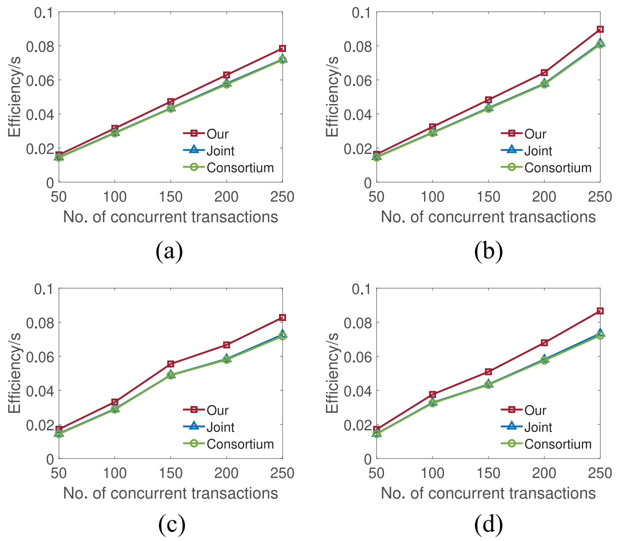
Fig. 8 plots the average time needed by the three consensus protocols in successfully generating a block. Among all the three consensus protocols, the consortium and joint consensus protocols deliver the best performance in terms of



(a) (b)

(c) (d)

Fig. 7. Accuracy comparison of different consensus protocol sets. (a) Accuracy comparison with $|F| = 50$. (b) Accuracy comparison with $|F| = 100$. (c) Accuracy comparison with $|F| = 150$. (d) Accuracy comparison with $|F| = 200$.



(a) (b)

(c) (d)

Fig. 8. Efficiency comparison of different consensus protocol sets. (a) Efficiency comparison with $|F| = 50$. (b) Efficiency comparison with $|F| = 100$. (c) Efficiency comparison with $|F| = 150$. (d) Efficiency comparison with $|F| = 200$.

block generation efficiency, where blocks are generated once all the followers and verifiers achieve consensus without any extra process. On the contrary, since our proposed public participation consensus adds a peer-prediction-based verification process to guarantee the trustworthiness of public verifiers and, further, the transaction verification accuracy, it needs more processing time.

VI. Conclusion

In this article, we proposed a novel public participation consortium blockchain system for infrastructure maintenance, which enables citizens to actively participate in the decision-making process, witnessing all administrative procedures in a real-time manner. To that aim, a compound blockchain architecture was introduced to achieve the goal of involving a verifier group, which is randomly and dynamically selected from the public citizens, to join in the transaction verification. Also, we designed a new consensus algorithm for this public participation consortium blockchain, which is compatible with the involvement of a dynamically extra verifier group in the blockchain. In particular, a private-prior peer-prediction-based truthful verification mechanism was devised to tackle the collusion attacks from public verifiers. Moreover, a cost-based incentive mechanism was designed to encourage the public to participate in the truthful verification process as much as possible. Simulation results indicated that the proposed truthful verification mechanism could achieve high accuracy and maintain efficiency.

VII. Discussion

A. Scalability Issue

For any blockchain-based system, the scalability issue should be considered carefully. In this article, we do not require all users who access our system to reach consensus. Basically, we only require a predefined group of committee members to achieve the consensus. Besides, we randomly choose a verifier group from public as representatives of citizens to participate in the transaction verification process for witnessing. Thus, we can cover more public users while maintaining the efficiency of the system.

The current application scenario of our system is for one city. However, if we broaden our scenario to cover a larger area, such as a state or province, or even the whole country, our system may be overwhelmed. In this scenario, we would require to use a special scalability solution for a blockchain-based system.

A sharding scheme would be helpful in this case as it overcomes the scalability challenge in blockchain [57]–[60]. The key idea of a sharding scheme in a blockchain is to partition the network into smaller committees, each of which processes a disjointed set of transactions (or a "shard"). Intuitively, it is helpful to improve the scalability by integrating the sharding scheme into our system. For example, in order to apply the sharding scheme to this larger scenario, we can divide the main blockchain for the state into several shards. Then, each shard committee can come to an agreement by using our consensus protocol.

Also, in this scenario, we must carefully consider several critical components, such as committee formation, cross-transaction processing, and epoch reconfiguration. Although we do not cover all of these components in this article, we would like to extend our scenario to states with consideration for scalability as a future work.

B. DoS Attacks

The Denial-of-Service (DoS) attack is a cyberattack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the

Internet. Our system needs a stable leader to provide liveness, and thus, DoS attacks targeting the leader can result in failure of the system. We can consider three DoS attack cases: 1) causing the leader to crash; 2) targeting nonleaders to make faulty nodes; and 3) performing burst attacks to disrupt the performance and stability of the network.

In the first case, the Raft algorithm has a mechanism to handle the leader crash. In this mechanism, the leader is required to send a "heartbeat" message at periodic intervals to all the followers. If a follower does not receive a heartbeat message form the leader within its leader election timeout interval, the leader is regarded as crashed, and the follower itself becomes a leader candidate, attempting to run for leader immediately.

In the second case, DoS attacks could also potentially target nonleader nodes to increase the number of faulty nodes. In the Raft algorithm, to invalidate the consensus protocol at least 50% faulty nodes in the network are required. However, all of these nodes, including the leader and nonleader, always have enough resources to deploy defense measures against DoS attacks. Thus, such attacks would be less practical than targeting the leader directly.

In the third case, attackers may stealthily launch burst DoS attacks at random intervals to affect the performance and stability of the network without crashing it. These attacks could also damage the Raft algorithm [61]. Specifically, Raft has three statically configured key parameters shared by all nodes, namely, heartbeat interval, minimum leader election timeout, and maximum leader election timeout. A key assumption underlying Raft is that the minimum leader election timeout should be greater than the broadcast time, which is defined as the average time of a node sending messages in parallel to every node in the network and receiving their responses. Hence, under the burst attacks, the broadcast time may repeatedly increase beyond any known bound, and then decrease back to normal levels, unpredictably and repeatedly violating Raft's underlying timing assumption. This can cause repeated leader elections. To prevent this attack, a babble-resistant Raft [61] can be deployed in our system.

References

[1] U. Habitat, *State of the World's Cities 2012/2013: Prosperity of Cities*. London, U.K.: Routledge, 2013.

[2] K. Davis, "The urbanization of the human population," *Sci. Amer.*, vol. 213, no. 3, pp. 40–53, 1965.

[3] R. Rivera, J. G. Robledo, V. M. Larios, and J. M. Avalos, "How digital identity on blockchain can contribute in a smart city environment," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, 2017, pp. 1–4.

[4] J. Markard, "Transformation of infrastructures: Sector characteristics and implications for fundamental change," *J. Infrastruct. Syst.*, vol. 17, no. 3, pp. 107–117, 2011.

[5] A. Ramaswami, A. G. Russell, P. J. Culligan, K. R. Sharma, and E. Kumar, "Meta-principles for developing smart, sustainable, and healthy cities," *Science*, vol. 352, no. 6288, pp. 940–943, 2016.

[6] M. Razaghi and M. Finger, "Smart governance for smart cities," *Proc. IEEE*, vol. 106, no. 4, pp. 680–689, Apr. 2018.

[7] H. Hou, "The application of blockchain technology in e-Government in China," in *Proc. IEEE 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2017, pp. 1–4.

[8] O. Konashevych, "The concept of the blockchain-based governing: Current issues and general vision," in *Proc. 17th Eur. Conf. Digit. Govt. (ECDG)*, 2017, p. 79.

[9] F. Tian, "An agri-food supply chain traceability system for china based on RFID & blockchain technology," in *Proc. IEEE 13th Int. Conf. Service Syst. Service Manag. (ICSSSM)*, 2016, pp. 1–6.

[10] Q. Lu and X. Xu, "Adaptable blockchain-based systems: A case study for product traceability," *IEEE Softw.*, vol. 34, no. 6, pp. 21–27, Nov./Dec. 2017.

[11] Z. Li, H. Wu, B. King, Z. B. Miled, J. Wassick, and J. Tazelaar, "On the integration of event-based and transaction-based architectures for supply chains," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, 2017, pp. 376–382.

[12] M. Nakasumi, "Information sharing for supply chain management based on block chain technology," in *Proc. IEEE 19th Conf. Bus. Informat. (CBI)*, vol. 1, 2017, pp. 140–149.

[13] Y. Madhwal and P. B. Panfilov, "Industrial case: Blockchain on aircraft's parts supply chain management," in *Proc. Amer. Conf. Inf. Syst. Workshop Smart Manuf.*, vol. 6, 2017, pp. 1–6.

[14] E. Yavuz, A. K. Koç, U. C. Çabuk, and G. Dalkılıç, "Towards secure e-voting using Ethereum blockchain," in *Proc. 6th Int. Symp. Digit. Forensic Security (ISDFS)*, 2018, pp. 1–7.

[15] G. Rowe and L. J. Frewer, "A typology of public engagement mechanisms," *Sci. Technol. Human Values*, vol. 30, no. 2, pp. 251–290, 2005.

[16] O. Coutard, *The Governance of Large Technical Systems*. London, U.K.: Routledge, 2002.

[17] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," in *Proc. 12th Annu. Int. Digit. Govt. Res. Conf. Digit. Govt. Innov. Challenging Times*, 2011, pp. 282–291.

[18] R. W. S. Ruhlandt, "The governance of smart cities: A systematic literature review," *Cities*, vol. 81, pp. 1–23, Nov. 2018.

[19] T. Dounas, D. Lombardi, and W. Jabi, "Framework for decentralised architectural design bim and blockchain integration," *Int. J. Archit. Comput.*, vol. 19, no. 2, pp. 157–173, 2020.

[20] D.-Y. Liao and X. Wang, "Applications of blockchain technology to logistics management in integrated casinos and entertainment," *Informatics*, vol. 5, no. 4, p. 44, 2018.

[21] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedREC: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, 2016, pp. 25–30.

[22] Z. Chen and Y. Zhu, "Personal archive service system using blockchain technology: Case study, promising and challenging," in *Proc. IEEE Int. Conf. AI Mobile Services (AIMS)*, 2017, pp. 93–99.

[23] Z. Yan, G. Gan, and K. Riad, "BC-PDS: Protecting privacy and self-sovereignty through blockchains for OpenPDS," in *Proc. IEEE Symp. Service Orient. Syst. Eng. (SOSE)*, 2017, pp. 138–144.

[24] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Security Privacy Workshops*, 2015, pp. 180–184.

[25] S. H. Hashemi, F. Faghri, and R. H. Campbell, "Decentralized user-centric access control using pubsub over blockchain," 2017. [Online]. Available: arXiv:1710.00110.

[26] S. H. Hashemi, F. Faghri, P. Rausch, and R. H. Campbell, "World of empowered IoT users," in *Proc. IEEE 1st Int. Conf. Internet Things Design Implement. (IoTDI)*, 2016, pp. 13–24.

[27] C. Liu, K. K. Chai, X. Zhang, E. T. Lau, and Y. Chen, "Adaptive blockchain-based electric vehicle participation scheme in smart grid platform," *IEEE Access*, vol. 6, pp. 25657–25665, 2018.

[28] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.

[29] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," in *Proc. Cloud Comput. Security Workshop (CCSW)*, 2017, pp. 45–50. [Online]. Available: https://doi.org/10.1145/3140649.3140656

[30] *Coinstack E-Voting Solution*. Accessed: Jan. 31, 2020. [Online]. Available: https://en.blocko.io/usecases/public/

[31] *Polys E-Voting System*. Accessed: Jan. 31, 2021. [Online]. Available: https://www.polys.me

[32] S. Nakamoto and A. Bitcoin. (2008). *A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[33] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *Proc. Annu. Cryptol. Conf.*, 2015, pp. 585–605. [Online]. Available: https://eprint.iacr.org/2013/796

[34] S. King and S. Nadal, "Ppcoin: peer-to-peer crypto-currency with proof-of-stake," self-published paper, vol. 19, no. 1, Aug. 2012.

[35] *Delegated Proof of Stake Explained*. Accessed: Jan. 31, 2021. [Online]. Available: https://academy.binance.com/en/articles/delegated-proof-of-stake-explained

[36] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, 2014.

[37] D. Mazieres, *The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus*, Stellar Develop. Found., San Francisco, CA, USA, 2015.

[38] S. Popov, "The tangle," Budapest, Hungary, IOTA, White Paper, 2018.

[39] D. Schwartz *et al.*, "The ripple protocol consensus algorithm," Ripple Labs, Inc., San Francisco, CA, USA, White Paper, 2014.

[40] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, vol. 99, pp. 173–186.

[41] *NEO White Paper*. Accessed: Jan. 31, 2021. [Online]. Available: https://docs.neo.org/docs/en-us/basic/whitepaper.html

[42] J. Kwon. (2014). *TenderMint: Consensus Without Mining*. [Online]. Available: http://jaekwon.com/2014/05/11/tendermint/

[43] G. D. Birkhoff, "Proof of the ergodic theorem," *Proc. Nat. Acad. Sci. USA*, vol. 17, no. 12, pp. 656–660, 1931. [Online]. Available: https://www.pnas.org/content/17/12/656

[44] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, Jun. 2014, pp. 305–319. [Online]. Available: https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro

[45] L. Lamport, "Paxos made simple," *ACM SIGACT News*, vol. 32, no. 4, pp. 18–25, 2001.

[46] A. A. Selcuk, E. Uzun, and M. R. Pariente, "A reputation-based trust management system for P2P networks," in *Proc. IEEE Int. Symp. Clust. Comput. Grid (CCGrid)*, 2004, pp. 251–258.

[47] Q. Zhuang, Y. Liu, L. Chen, and Z. Ai, "Proof of reputation: A reputation-based consensus protocol for blockchain based systems," in *Proc. Int. Electron. Commun. Conf.*, 2019, pp. 131–138.

[48] J. Yu, D. Kozhaya, J. Decouchant, and P. Esteves-Verissimo, "RepuCoin: Your reputation is your power," *IEEE Trans. Comput.*, vol. 68, no. 8, pp. 1225–1237, Aug. 2019.

[49] G. Wang, "RepShard: Reputation-based sharding scheme achieves linearly scaling efficiency and security simultaneously," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, 2020, pp. 237–246.

[50] C. Huang *et al.*, "RepChain: A reputation based secure, fast and high incentive blockchain system via sharding," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4291–4304, Mar. 2021.

[51] J. Xie *et al.*, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.

[52] J. Witkowski and D. C. Parkes, "Peer prediction without a common prior," in *Proc. 13th ACM Conf. Electron. Commerce (EC)*, 2012, pp. 964–981. [Online]. Available: https://doi.org/10.1145/2229012.2229085

[53] N. Miller, P. Resnick, and R. Zeckhauser, "Eliciting informative feedback: The peer-prediction method," *Manag. Sci.*, vol. 51, no. 9, pp. 1359–1373, 2005. [Online]. Available: https://doi.org/10.1287/mnsc.1050.0379

[54] J. Emile, "Stackelberg (Heinrich von)—The theory of the market economy, translated from the German and with an introduction by Alan T. PEACOCK," *Revue Aconomique*, vol. 4, no. 6, pp. 944–945, 1953.

[55] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 35–47, Jan. 2018.

[56] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[57] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "SOK: Sharding on blockchain," in *Proc. 1st ACM Conf. Adv. Financ. Technol.*, 2019, pp. 41–61.

[58] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proc. Int. Conf. Manag. Data*, 2019, pp. 123–140.

[59] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 17–30.

[60] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor, "Blockchain and scalability," in *Proc. IEEE Int. Conf. Softw. Qual. Rel. Security Companion (QRS-C)*, 2018, pp. 122–128.

[61] R. Hanmer, S. Liu, L. Jagadeesan, and M. R. Rahman, "Death by babble: Security and fault tolerance of distributed consensus in high-availability softwarized networks," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, 2019, pp. 266–270.

Yuhao Bai (Graduate Student Member, IEEE) received the B.S. degree from the Department of Mathematics and Applied Mathematics, Harbin Institute of Technology, Weihai, China, in 2018. He is currently pursuing the Ph.D. degree in electronic and electrical engineering with Hanyang University, Seoul, South Korea.

His research interests include blockchain, privacy protection, and post-quantum cryptography.

Qin Hu received the Ph.D. degree in computer science from George Washington University, Washington, DC, USA, in 2019.

She is currently an Assistant Professor with the Department of Computer and Information Science, Indiana University–Purdue University Indianapolis, Indianapolis, IN, USA. Her research interests include wireless and mobile security, mobile-edge computing, blockchain, and crowdsourcing/crowdsensing.

Seung-Hyun Seo (Member, IEEE) received the B.S. degree from the Department of Mathematics, Ewha Womans University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in computer science from Ewha Womans University, Seoul, in 2002 and 2006, respectively.

She is currently a Professor with Hanyang University, Seoul. Before joining as a Faculty with Hanyang University, in 2017, she was an Assistant Professor with Korea University Sejong Campus, Yeongi-gun, South Korea, for two years. Before that, she was a Postdoctoral Researcher of Computer Science with Purdue University, West Lafayette, IN, USA, for two and half years, a Senior Researcher with Korea Internet and Security Agency, Seoul, for two years, and a Researcher for three years with Financial Security Agency, Seoul. Her main research interests include cryptography, IoT security, mobile security, blockchain, and post-quantum cryptography.

Kyubyung Kang (Associate Member, IEEE) received the Ph.D. degree in civil engineering from Purdue University, West Lafayette, IN, USA, in 2018.

He is an Assistant Professor of Construction Management with the Department of Engineering Technology, Indiana University–Purdue University Indianapolis, Indianapolis, IN, USA. His research interests are smart infrastructure management, machine learning, and 3-D modeling.

John J. Lee (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical and computer engineering (ECE) from the Georgia Institute of Technology, Washington, DC, USA, in 2003 and 2004, respectively.

He joined as a Faculty with the Department of ECE, Indiana University–Purdue University Indianapolis, Indianapolis, IN, USA, in 2005, where he is currently an Associate Professor. He has published eight SCI-indexed and nine SCIE-indexed journals among others. His research interests include IoT, blockchain technology, parallel acceleration of algorithms and applications, and high-performance computing.