PAPER

Improved PBFT-Based High Security and Large Throughput Data Resource Sharing for Distribution Power Grid

Zhimin SHAO[†], **Chunxiu LIU**[††a)], **Cong WANG**[††], **Longtan LI**[†††], **Yimin LIU**[††], ***and*** **Zaiyan ZHOU**[††], ***Nonmembers***

SUMMARY Data resource sharing can guarantee the reliable and safe operation of distribution power grid. However, it faces the challenges of low security and high delay in the sharing process. Consortium blockchain can ensure the security and efficiency of data resource sharing, but it still faces problems such as arbitrary master node selection and high consensus delay. In this paper, we propose an improved practical Byzantine fault tolerance (PBFT) consensus algorithm based on intelligent consensus node selection to realize high-security and real-time data resource sharing for distribution power grid. Firstly, a blockchain-based data resource sharing model is constructed to realize secure data resource storage by combining the consortium blockchain and interplanetary file system (IPFS). Then, the improved PBFT consensus algorithm is proposed to optimize the consensus node selection based on the upper confidence bound of node performance. It prevents Byzantine nodes from participating in the consensus process, reduces the consensus delay, and improves the security of data resource sharing. The simulation results verify the effectiveness of the proposed algorithm.

key words: *distribution power grid, practical byzantine fault tolerance, data resource sharing, node trust degree, upper confidence bound*

1. Introduction

Distribution power grid plays an important role in connecting distributed renewable generators, loads, and storages with the grid [1]–[4]. To ensure its reliable and safe operation, various types of data including infrastructure data, metering data, topology data, and graphic data, are intensively collected for real-time state monitoring and deep analysis [5], [6]. However, these data are distributed in different systems and departments, which leads to information islands [7], [8]. Moreover, it is difficult to realize collaborative data analysis and improve data utilization efficiency due to the lack of secure data resource sharing mechanism [9]–[11]. Data resource sharing faces several security issues such as malicious attacks and data tempering. These security issues will result in privacy breaches of grid users, sensitive information leakage, and even lead to grid operation failure [12]–[14]. Therefore, how to realize high-security and real-time data resource sharing for distribution power grid has become an important research topic.

Consortium blockchain is controlled and managed by a consortium of multiple departments. It utilizes encryption techniques for data validation and storage, while employing consensus algorithms for block generation and updates [15]–[19]. The consortium chain has a dedicated identity and permission management system, which can isolate the blockchain of different departments while retaining the access permissions of the consortium members. Consortium blockchain not only ensures the security and efficiency of multi-department data sharing, but also has a high tamper resistance to ensure data integrity and traceability. The core of consortium blockchain relies on consensus, which allows untrusted blockchain nodes to achieve data consistency through predetermined consensus mechanism [20], [21]. The reliability of efficiency of consensus algorithm play a vital role in ensuring high-security and real-time data resource sharing for distribution power grid. Among various consensus algorithms, the practical Byzantine fault tolerance (PBFT) algorithm provides safeguard against Byzantine nodes based on the principles of majority agreement and collaborative decision making [22]–[24]. It possesses advantages of high security and low reward variance, making it suitable for data resource sharing in distribution power grid. Several studies have investigated on PBFT-based consensus in consortium blockchain. In [25], Abishu et al. applied PBFT consensus algorithm in consortium blockchain-enabled energy trading based on wireless power transfer to ensure the security and privacy of transactions between untrustworthy electric vehicles. In [26], Zhang et al. proposed antiquantum privacy protection scheme of smart meter based on consortium blockchain. A low-communication cost PBFT consensus algorithm was developed to prevent user privacy leakage and tampering of power consumption data. However, despite its great potential technical advantages, the direct application of PBFT on data resource sharing in distribution power grid still faces several key challenges.

First, conventional PBFT algorithm involves the vast majority of nodes in consensus. It cannot work efficiently in distribution power grid with large number of nodes. Both communication overhead and computation complexity grow exponentially with node scale, which results in unbearable consensus delay. Second, it is susceptible to Sybil attacks due to the arbitrary selection of consensus nodes. Sybil attacks on the power grid pose risks such as control system manipulation, data integrity compromise, and potential cas-

Manuscript received November 22, 2023.
Manuscript revised December 25, 2023.
Manuscript publicized January 31, 2024.

[†]State Grid Shandong Electric Power Company, Jinan, 250001, China.
[††]State Grid Shandong Dezhou Power Supply Company, Dezhou, 253057, China.
[†††]State Grid Shandong Pingyuan Power Supply Company, Dezhou, 253100, China.
a) E-mail: liu_chunxiu2023@163.com

DOI: 10.1587/transfun.2023EAP1150

cading outages due to the interconnected nature of the grid. Existing node selection approaches such as trust-PBFT [27], two-stage PBFT [28] and game-based PBFT [29] mainly focus on recent node trust degree, but ignores performances of historical trust degree and consensus delay. Last not but least, confidence bound of node trust degree has been largely neglected in existing node evaluation and selection mechanisms. Some normal nodes are misjudged as Byzantine nodes due to the fluctuations of communication and computing resources, which significantly reduces consensus reliability and increases consensus delay.

Several studies have been conducted to explore the application of consortium blockchain with PBFT-based consensus in data resource sharing. In [30], a patient-controlled electronic health records sharing scheme based on cloud computing collaborating consortium blockchain technology was proposed, and a node-state-checkable PBFT consensus algorithm was applied to reduce the impact of the malicious node on the whole consortium blockchain. In [31], a consortium blockchain-based secure data sharing scheme of internet of vehicles was proposed to implement automatic registration, rapid authentication, and reliable sharing. The PBFT consensus algorithm was adopted to ensure the consistency of the entire network ledger. The PBFT-based consensus mechanisms utilized in these works require the participation of the vast majority of nodes on consensus, which is not suitable for real-world application in large-scale data resource sharing in distribution power gird. Some researchers investigate node selection in PBFT to address scalability issues. In [27], Tong et al. proposed a trust-PBFT consensus algorithm to improve the fault tolerance performance and scalability, which introduced the peer-to-peer trust calculation model to evaluate the trust degree of nodes that qualify as participants of PBFT. In [28], Qushtom et al. proposed a consensus mechanism that integrates proof-of-stake with PBFT, which can effectively deal with dishonest nodes and maintain high performance by using node trust degree and reward mechanisms as crucial components of the block validation and ordering processes. However, these works lack accurate evaluation of node trust degree based on both historical and recent performances, which causes high Byzantine node ratio and frequent security breaches. In [32], Gao et al. proposed a novel optimized PBFT consensus algorithm based on EigenTrust model, which obtained a unique trust value for every node in the system by recording the transaction history between nodes. In [33], Xiang et al. proposed a distributed PBFT consensus algorithm suitable for virtual power plant transaction blockchain to meet the requirements of privacy and efficiency of power data consensus, which designed the credit evaluation indicators according to the historical performance evaluation coefficients and recent debt evaluation coefficients of the blockchain nodes. These works ignored the paradox relationship between reliable node selection and consensus delay. The reduction of Byzantine node ratio based on complicated node trust degree evaluation and selection approaches is at the cost of high consensus delay. It is intuitive to achieved balanced performance between security and efficiency by jointly considering trust degree and consensus delay. Furthermore, the influence of fluctuation of communication and computing resources on node evaluation and selection have not been investigated.

To tackle these challenges, we propose an improved PBFT consensus algorithm based on intelligent consensus node selection to realize high-security and real-time data resource sharing for distribution power grid. First, we construct a consortium blockchain-based data resource sharing model for distribution power grid and elaborate the design of secure data storage and smart contract mechanisms. Then, the improved PBFT consensus algorithm based on intelligent consensus node selection is proposed. It learns to optimize consensus node selection based on upper confidence bound to prevent Byzantine nodes from participating in the consensus process. Finally, the security and delay performances of the proposed algorithm are verified by simulations. The contributions of this paper are summarized as follows.

- *Consortium blockchain-based data resource sharing framework for distribution power grid:* We propose a high-security and low-delay data resource sharing framework by combining consortium blockchain with interplanetary file system (IPFS). We develop a secure resource storage mechanism to ensure data immutability and reduce consensus delay. The data resources are encrypted and stored in IPFS while only the Hash addresses as transaction attributes are stored in blockchain. We also design advanced storage contract and query contract to facilitate transparent and automatic execution of data resources.
- *Improved PBFT consensus based on intelligent consensus node selection:* We construct upper confidence bound of node performance to optimize consensus node selection. The upper confidence bound achieves a balanced tradeoff between exploration of potential consensus nodes and exploitation of nodes with better empirical performances. We further consider consensus delay, historical node trust degree, and recent node trust degree in empirical performance evaluation.
- *Extensive performance analysis on consensus delay, throughput, and security:* We conduct extensive simulations on consensus delay, throughput, and security to validate the effectiveness of the proposed algorithm. We consider various performance metrics including delay, transactions per second (TPS), and Byzantine node ratio. We also analyze its vulnerability to replay attacks, distributed denial of service (DDoS) attacks, and Sybil attacks, and provide simulation verification in terms of probability of successful attack.

This paper is organized as follows. Section 2 introduces the consortium blockchain-based data resource sharing model for distribution power grid. Section 3 proposes the improved PBFT consensus algorithm based on intelligent consensus node selection. Simulation results are given in Sect. 4. Section 5 presents conclusion.

2. Consortium Blockchain-Based Data Resource Sharing Model for Distribution Power Grid

The consortium blockchain-based data resource sharing model for distribution power grid is shown in Fig. 1, which includes the data acquisition system, client, data resource sharing consortium blockchain and IPFS. The details are described as follows.

- **Data acquisition system:** Data acquisition systems of distribution power grid mainly include supervisory control and data acquisition system (SCADA), power production management system (PMS), energy management system (EMS), advanced metering infrastructure (AMI), geographic information system (GIS), and etc, which collect different types data, such as voltage, current, harmonic wave, temperature, and weather. These data are first encrypted according to the random symmetric key generated by the data acquisition system to obtain the encrypted file. The data acquisition system then uploads the generated encrypted file to IPFS for storage according to the smart contract. Smart contracts are automated contracts based on blockchain technology, and they are capable of executing, managing, and enforcing the terms of a contract without a third party.
- **Client:** Clients include personnel from departments such as dispatching, operation and detection, and marketing. After completing identity authentication, client sends a query request of data resources to the blockchain based on the Hash address. The client calls the corresponding smart contract to achieve secure cross-department sharing of data resources to realize refined management of distribution power grid.
- **Data resource sharing consortium blockchain:** The blockchain includes the blockchain node network, the smart contract and the consensus algorithm. The blockchain node network is composed of N nodes, the set of which is represented as $\mathcal{V} = \{V_1, V_2, \ldots, V_n, \ldots, V_N\}$. The smart contract is accessed and called by data acquisition systems and clients to ensure consistency and transparency in the data resource sharing process. The consensus algorithm serves as the governing rule followed by all blockchain nodes to ensure data security. The data resource sharing consortium blockchain uses a special identity and permission management system to securely store the data resources of a certain department while retaining the access rights of other departments, which ensures the security of cross-department sharing of data resources. Besides, it has a high tamper resistance to ensure data integrity and traceability.
- **IPFS:** IPFS is an open source distributed file system. Compared with conventional centralized file storage systems, which can only rely on servers to download files, IPFS runs on peer-to-peer networks, avoiding the single point failure and vulnerability of conventional centralized network. In addition, compared with the conventional location-based addressing method which is easy to cause the loss of data resources, IPFS defines a distribution protocol based on content addressing. Each file has a unique Hash address, which not only has the tamper resistance function, but also can prevent uploading a large number of duplicate data.

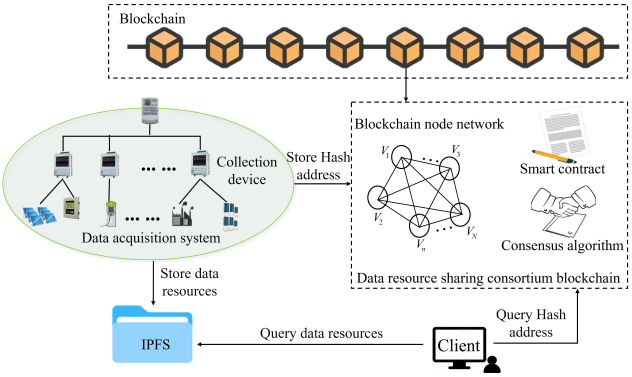


Fig. 1 The consortium blockchain-based data resource sharing model for distribution power grid.

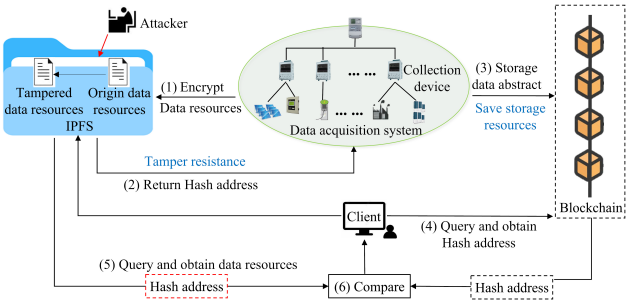


Fig. 2 The secure data resource storage mechanism.

In this paper, T iterations are considered, and the set is represented as $\mathcal{T} = \{1, 2, \ldots, t, \ldots, T\}$. In each iteration, firstly, according to secure data resource storage mechanism, data acquisition system uploads collected data resources to IPFS, and uploads the corresponding data abstract to the blockchain. Secondly, the blockchain generates and stores data resource identity document (DRID) in a certain block according to the data resource storage contract. Finally, the client query data resources according to data resource query contract. The details are expressed as follows.

2.1 Secure Data Resource Storage Mechanism

Each block in the blockchain can only store a limited amount of data. If the data resources from the data acquisition system are stored directly in the blockchain, a large number of blocks will be required, resulting in increased consensus delay when the client queries the data resources. We propose a secure and real-time data resource storage mechanism by combining blockchain and IPFS, which is shown in Fig. 2. The process is described as follows.

(1) Data acquisition system encrypts the data resources and uploads it to IPFS according to smart storage contract for storage.

(2) When receiving the data resources, IPFS calculates the Hash address through Hash operation and sends the Hash address back to data acquisition system.

(3) Data acquisition system constructs a data abstract according to the received Hash address and uploads it to the blockchain according to the smart contract. The blockchain generates DRID and stores it in a certain block.

(4) When a client queries the data resources, it sends a request to the blockchain and obtains the corresponding Hash address in DRID according to the smart contract.

(5) Based on the received Hash address, the client sends a query request to IPFS and retrieves the corresponding data resources.

(6) When receiving the data resources, the client decrypts it and calculates the Hash address. By comparing this Hash address with the one stored in blockchain, the client can guarantee the security of the data resources sharing. If the two Hash addresses are consistent, the client receives the correct data resources. Otherwise, the client receives the tampered data resources.

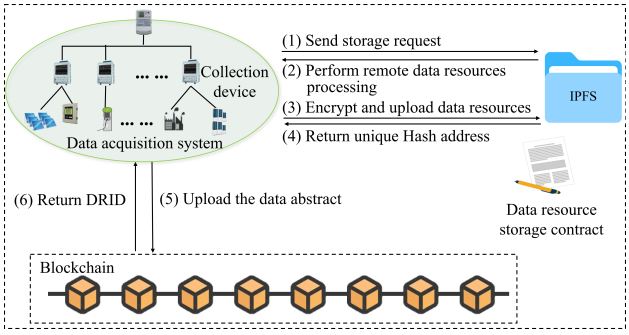
In terms of real-time performance, since the amount of data abstract generated based on Hash address is small, storing data abstract in consortium blockchain instead of data resources can effectively save storage resources on consortium blockchain, and improve the efficiency of consensus algorithms. In terms of security performance, any attack on the data resources will change the Hash address. As the Hash address stored in the consortium blockchain is immutable, comparing the two Hash addresses ensures the immutability and traceability of data resources.

2.2 Smart Contract Design

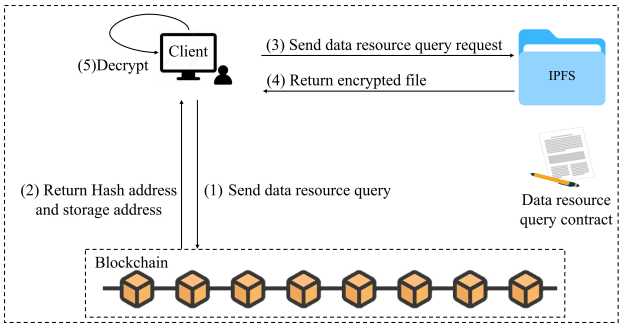
Smart contracts are programs that are stored on consortium blockchain and executed automatically when predefined conditions are met. Smart contracts support automated execution, immutability of data and decentralization of node network. Smart contracts are used to solve the problems of non-transparent execution process and non-uniform execution standards which are encountered in data resource sharing. The core of the smart contracts consists of data resource storage contract and data resource query contract.

2.2.1 Data Resource Storage Contract

Data resource storage contract is used to store the data resources in the IPFS and store the data abstracts in consortium blockchain. The execution of the data resource storage contract is shown in Fig. 3(a), which is described as follows.



(a) Data resource storage contract



(b) Data resource query contract

Fig. 3 The smart contact design: (a) data resource storage contract; (b) data resource query contract.

(1) Data acquisition system sends the data resource storage request to IPFS.

(2) When receiving the request, IPFS performs remote data resources processing for the system according to remote procedure call (RPC) protocol.

(3) Following the RPC protocol, the system encrypts the data resources and uploads them to IPFS. First, the system generates a random symmetric key Key according to the key generation operation $KeyGen(.)$, which is given by

$$Key = KeyGen(\lambda) \tag{1}$$

where λ is the security parameter. Then, the system encrypts data resource $DataRes$ using the generated random symmetric key Key, executes encryption operation $Encrypt_{sse}(.)$, and obtains output encrypted file C_{ds}, which is given by

$$C_{ds} = Encrypt_{sse}(Key, DataRes) \tag{2}$$

Finally, the system uploads C_{ds} to IPFS.

(4) When receiving the data resources, IPFS creates storage addresses $IPFS_{add}$ for data resources, calculates the Hash address $Hash_{file}$ of the data resources, and returns it to the data acquisition system.

(5) The system constructs the data abstract according to the received Hash address and uploads it to blockchain.

(6) When receiving the data abstract, the blockchain combines it with system ID ID_{sys}, the random symmetric key Key for encryption, and timestamp TP_{sto} for storage to generate a DRID, and stores DRID in a new block. Then, the DRID is return to the system. DRID is expressed as

$$DRID =< ID_{sys}, Key, Hash_{file}, IPFS_{add}, TP_{sto} > \quad (3)$$

2.2.2 Data Resource Query Contract

Data resource query contract is used to provide data resource query service for the client. The execution of data resource query contract is shown in the Fig. 3(b), which is described as follows.

(1) The client sends the data resource query request $REQUEST$ to the blockchain, which is given by

$$REQUEST =< DRID, TP_{que}, ID_{cli}, Sig_{cli} > \quad (4)$$

where TP_{que} is the timestamp for query, ID_{cli} and Sig_{cli} are the client ID and client signature, respectively.

(2) When receiving the query request $REQUEST$, blockchain searches for the corresponding data abstract based on the DRID. Subsequently, blockchain returns the corresponding Hash address $Hash_{file}$ and storage addresses $IPFS_{add}$ to the client.

(3) Based on the received $Hash_{file}$ and $IPFS_{add}$, the client sends a data resource query request to IPFS.

(4) When receiving the query request, IPFS retrieves and sends the encrypted file C_{ds} to the client.

(5) When receiving the encrypted file C_{ds}, the client verifies its Hash address. If the verification passes, the client uses key Key to decrypt C_{ds} according to decryption operation $Decrypt_{sse}(.)$, and obtains the data resources $DataRes$, which is given by

$$DataRes = Decrypt_{sse}(Key, C_{ds}) \quad (5)$$

3. Improved PBFT Consensus Algorithm Based on Intelligent Consensus Node Selection

In the data resource sharing process, the consortium blockchain utilizes the PBFT consensus algorithm to authenticate the participating nodes. PBFT consensus algorithm is a widely adopted consensus algorithm in consortium blockchains and is well known for its effectiveness in solving the Byzantine problem. PBFT consensus algorithm involves client and consensus nodes. The client sends the consensus request to realize the data resource sharing. The consensus nodes refer to the blockchain nodes that participate in the consensus process, which can be further divided into a master node and several slave nodes. The master node is randomly selected, which is responsible for receiving consensus request from client and broadcasting it to slave nodes. The slave nodes are the remaining consensus nodes, which are responsible for validating and processing received consensus requests. PBFT consensus algorithm follows the principle of majority rule, and consensus is reached when the consensus request sent by client is confirmed by more than half of the consensus nodes. This means that each consensus node needs to communicate with each other, resulting in significant communication overhead. During the consensus process, consensus node may encounter situations where confirmation messages are lost due to the fluctuations of communication and computing resources, or confirmation messages are tampered with due to attack. Such consensus node is called Byzantine node. Conversely, consensus node that sends the correct confirmation message is called non-Byzantine node. If a Byzantine node participates in the consensus process as a master node, the client will not receive any correct confirmation messages. The above features lead to some limitations of the conventional PBFT consensus algorithm, which are described as follows.

- PBFT consensus algorithm requires all blockchain nodes to participate in the consensus process as consensus nodes. However, the characteristic that each consensus node needs to communicate with each other determines that the PBFT consensus algorithm has significant communication overhead. As the number of blockchain nodes increases, the communication overhead increases rapidly, resulting in an increased consensus delay.
- Since the PBFT consensus algorithm follows the principle of majority rule, it is necessary to ensure that the number of non-Byzantine nodes is greater than the number of Byzantine nodes. However, the PBFT consensus algorithm does not have the ability to identify Byzantine nodes. When a Byzantine node serves as the master node to participate in consensus process, or when a large number of Byzantine nodes participate in consensus process, the consensus process will fail. The client needs to reinitiate the consensus. This leads to a serious increase in consensus delay and a decrease in the security of data resource sharing.

To address the above challenges, we propose the improved PBFT consensus algorithm based on intelligent consensus node selection. Firstly, the consensus delay model, node trust degree model and consensus node selection approach based on upper confidence bound are proposed to improve the PBFT consensus algorithm. Then the implementation of the improved PBFT consensus algorithm is described. The flow of the improved PBFT consensus algorithm is shown in Fig. 4.

3.1 Consensus Delay Model

In the improved PBFT consensus algorithm, the client sends the consensus request to the master node, and the master node

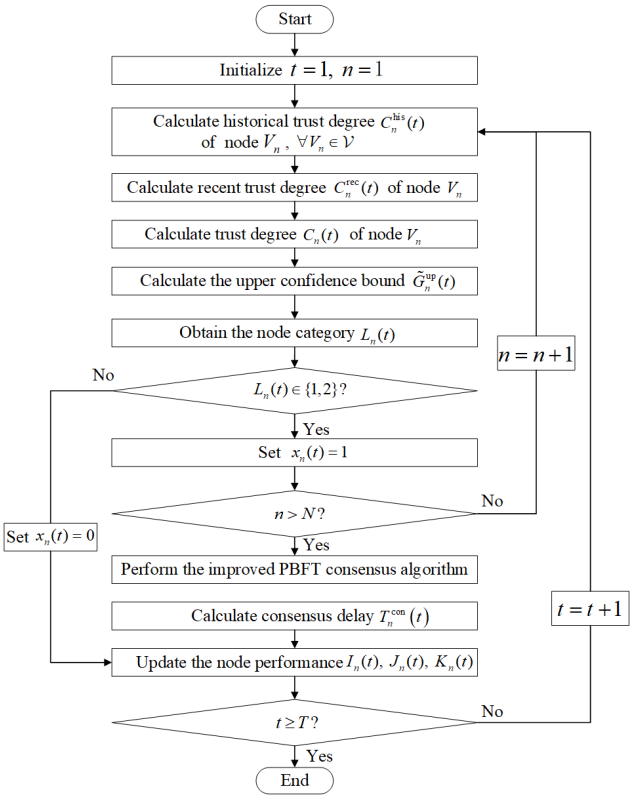


Fig. 4 The flow of the improved PBFT consensus algorithm.

broadcasts it to the slave nodes. Then, consensus nodes validate the consensus request and reply to the client. Define the consensus delay as the time difference between the time when the client sends a consensus request to master node and the time when the client receives a reply from the consensus node. Consensus delay is a key indicator to measure the performance of consensus algorithms, which is affected by the communication resources between client and consensus nodes, the computing resources of consensus nodes, and the number of consensus nodes [34]. Assuming that at time $\tau^{\text{req}}(t)$ in the t-th iteration, the client sends a consensus request to the master node, and at time $\tau_n^{\text{rep}}(t)$, the client receives the reply from node V_n. Denote the consensus delay of node V_n in the t-th iteration as $T_n^{\text{con}}(t)$, which is given by

$$T_n^{\text{con}}(t) = \tau_n^{\text{rep}}(t) - \tau^{\text{req}}(t) \tag{6}$$

3.2 Node Trust Degree Model

The node historical trust degree considers the performance of nodes in past interactions, revealing their stability, consistency, and honesty. If a node has been performing well in the past, it is possible to give it a higher trust degree. The node recent trust degree takes into account the recent behavior of nodes, as their behavior may change over time. Even if a node has performed well in the past, its recent trust degree may be low if its recent behavior is unreliable. The node historical trust degree mainly focuses on the long-term performance of nodes, while the node recent trust degree focuses on the instantaneous performance of nodes. Therefore, if the behavior of a node changes in a short period of time, the node recent trust degree may better reflect the actual situation. Considering both can help establish a more comprehensive trust model.

3.2.1 Node Historical Trust Degree Model

The node historical trust degree is calculated based on the consensus process before current iteration. At the beginning of the t-th iteration, define $K_n(t)$ as the number of times that node V_n has served as consensus node, define $I_n(t)$ as the number of times that node V_n has served as non-Byzantine node, and define $J_n(t)$ as the number of times that node V_n has served as Byzantine node. The node historical trust degree of node V_n in the t-th iteration is denoted as

$$C_n^{\text{his}}(t) = \begin{cases} \frac{I_n(t)-\alpha J_n(t)}{K_n(t)}, & \text{if } K_n(t) > 0 \\ 0, & \text{if } K_n(t) = 0 \end{cases} \tag{7}$$

where $\alpha > 1$ is the historical trust degree adjustment coefficient to amplify the effect of the number of times that node has been Byzantine node. With the adjustment of α, the node historical trust degree decreases rapidly with the number of times that node has been Byzantine node, which can effectively improve the security of data resource sharing. In particular, set the initial node historical trust degree as $C_n^{\text{his}}(1) = 0$.

3.2.2 Node Recent Trust Degree Model

The node recent trust degree is calculated based on the node behavior in the last iteration. The node behavior includes normal behavior, fault behavior and malicious behavior. Normal behavior refers that nodes send and receive messages as specified. Fault behavior refers that nodes cannot send messages or send error messages due to operational failure. Malicious behavior refers that nodes send inconsistent messages intentionally. In particular, non-Byzantine nodes perform normal behavior, and Byzantine nodes perform fault behavior and malicious behavior.

Define $A_n^{\text{nor}}(t) \in \{0,1\}$ to describe whether node V_n performs normal behavior in the t-th iteration, where $A_n^{\text{nor}}(t) = 1$ means that node V_n performs normal behavior, and $A_n^{\text{nor}}(t) = 0$ otherwise. Similarly, define $A_n^{\text{fau}}(t) \in \{0,1\}$ to describe whether node V_n performs fault behavior, and define $A_n^{\text{mal}}(t) \in \{0,1\}$ to describe whether node V_n performs malicious behavior. The recent trust degree of node V_n in the t-th iteration is denoted as

$$C_n^{\text{rec}}(t) = A_n^{\text{nor}}(t-1) + \beta^2 A_n^{\text{fau}}(t-1) + \beta^4 A_n^{\text{mal}}(t-1) \tag{8}$$

where $\beta \geq 1$ is the recent trust degree adjustment coefficient. In particular, set the initial node recent trust degree as $C_n^{\text{rec}}(1) = 0$.

3.2.3 Node Trust Degree Model

Denotes the initial trust degree of node V_n as C_n^{ini}, which is determined by node type, node computing resource and communication capability. The node trust degree is given by

$$C_n(t) = C_n^{\text{ini}} + \gamma_h C_n^{\text{his}}(t) + \gamma_r C_n^{\text{rec}}(t) \tag{9}$$

where γ_h and γ_r are the historical trust degree weight and the recent trust degree weight, respectively. By dynamically adjusting the settings of γ_h and γ_r, the importance of the node historical trust degree and node recent trust degree can be changed.

3.3 Intelligent Consensus Node Selection Based on Upper Confidence Bound

Define the consensus node selection variable as $x_n(t) \in \{0, 1\}$, where $x_n = 1$ means node V_n is selected to participate in consensus process at the t-th iteration, and $x_n(t) = 0$ otherwise. An intelligent consensus node selection based on upper confidence bound is proposed to realize the classification and selection of nodes. This approach can accurately identify Byzantine nodes and prevent them from participating in the consensus process to reduce the number of consensus nodes. The proposed algorithm is described as follows.

3.3.1 Node Performance Evaluation

In order to achieve accurate evaluation of node performance, we comprehensively consider the influence of consensus delay and node trust degree. Define the performance of node V_n in the t-th iteration as weighted sum of consensus delay $T_n^{\text{con}}(t)$ and node trust degree $C_n(t)$, which is given by

$$G_n(t) = T_n^{\text{con}}(t) + \mu C_n(t) \tag{10}$$

where μ is the weight of node trust degree.

In the consensus process, due to the large number of blockchain nodes and complex communication environment, it is difficult to obtain the global information of the current iteration, which makes it difficult to directly select consensus nodes based on the node performance evaluation described in (10). In addition, the fluctuations of computing resources and communication resources will lead to misjudgments of Byzantine nodes. To solve the above problems, we develop node performance evaluation based on the upper confidence bound. The upper confidence bound of node V_n in the t-th iteration is given by

$$\tilde{G}_n^{\text{up}}(t) = \bar{G}_n(t) + \theta\sqrt{\frac{\ln t}{K_n(t)}} \tag{11}$$

where θ is the adjustment coefficient. $\bar{G}_n(t)$ is the historical average performance of node V_n in the previous $t-1$ iterations, which is given by

Table 1 The relationship between blockchain node category and the upper confidence bound.

Blockchain node category	$L_n(t)$	Condition
Optimal node	1	The node satisfies condition $\tilde{G}_n^{\text{up}}(t) \geq \tilde{G}_n^{\text{tru}}$ and has the largest $\tilde{G}_n^{\text{up}}(t)$
Trusted node	2	The node satisfies condition $\tilde{G}_n^{\text{up}}(t) \geq \tilde{G}_n^{\text{tru}}$ and dose not have the largest $\tilde{G}_n^{\text{up}}(t)$
Alternate node	3	The node satisfies condition $\tilde{G}_n^{\text{up}}(t) < \tilde{G}_n^{\text{tru}}$

$$\bar{G}_n(t) = \frac{1}{K_n(t)} \sum_{t_0=1}^{t-1} x_n(t_0) G_n(t_0) \tag{12}$$

According to (11), the improved PBFT consensus algorithm exploits nodes according to historical average performance, and explores node performance according to confidence radius. Besides, it uses adjustment coefficient to balance the exploitation and exploration. By introducing the upper confidence bound, the proposed algorithm can select consensus nodes with good performance to participate in the consensus process and prevent Byzantine nodes from disturbing the consensus process, which effectively improves the security of data resource sharing and reduces the consensus delay.

3.3.2 Consensus Node Selection

The improved PBFT consensus algorithm divides blockchain nodes into three categories based on the upper confidence bound, which are called optimal node, trusted node and alternate node. Define the category of node V_n as $L_n(t) \in \{1, 2, 3\}$, where $L_n(t) = 1$ represents the optimal node, $L_n(t) = 2$ represents the trusted node and $L_n(t) = 3$ represents the alternate node. Define $\tilde{G}_n^{\text{tru}}$ as the threshold of trusted node. The relationship between blockchain node category and the upper confidence bound is shown in Table 1.

In the improved PBFT consensus algorithm, the optimal node participates in the consensus process as the master node, the trusted nodes participate in the consensus process as the slave node, and the alternate nodes do not participate in the consensus process due to the poor performance. That is to say, node V_n participates in the consensus process when $L_n(t) \in \{1, 2\}$, while node V_n does not participate in the consensus process when $L_n(t) = 3$. Therefore, the consensus node selection is given by

$$x_n(t) = \begin{cases} 1, & L_n(t) \in \{1, 2\} \\ 0, & L_n(t) = 3 \end{cases} \tag{13}$$

3.3.3 Node Performance Update

When the consensus node selection in the t-th iteration is decided, the client performs the consensus process according to the improved PBFT consensus algorithm based on intelligent consensus node selection. After the consensus process is complete, the proposed algorithm analyzes the behaviors of the consensus nodes. If consensus node V_n performs normal behavior, set $A_n^{\text{nor}}(t) = 1$, $A_n^{\text{fau}}(t) = 0$ and $A_n^{\text{mal}}(t) = 0$. If

consensus node V_n performs fault behavior, set $A_n^{\text{fau}}(t) = 1$, $A_n^{\text{nor}}(t) = 0$ and $A_n^{\text{mal}}(t) = 0$. If consensus node V_n performs malicious behavior, set $A_n^{\text{mal}}(t) = 1$, $A_n^{\text{nor}}(t) = 0$ and $A_n^{\text{fau}}(t) = 0$. Then, update the node performance as

$$I_n(t+1) = I_n(t) + A_n^{\text{nor}}(t) \tag{14}$$

$$J_n(t+1) = J_n(t) + A_n^{\text{fau}}(t) + A_n^{\text{mal}}(t) \tag{15}$$

$$K_n(t+1) = K_n(t) + x_n(t) \tag{16}$$

3.4 Implementation of the Improved PBFT Consensus Algorithm

The consensus process is performed according to the selected consensus nodes based on the improved PBFT consensus algorithm. An implementation of improved PBFT consensus algorithm is shown in Fig. 5. There are $N = 6$ blockchain nodes, consisting of 1 optimal node, 4 trusted nodes, and 1 alternate node. The optimal node is V_3, the trusted nodes are V_1, V_2, V_4 and V_5, and the alternate node is V_6. It should be noted that the alternate node V_6 is a Byzantine node, which is accurately identified by the improved PBFT consensus algorithm and is prevented from participating in the consensus process.

The consensus process based on improved PBFT consensus algorithm includes request, pre-prepare, prepare, commit and reply, which are described as follows.

- **Request:** The client verifies the data format and digital signature to construct request message and ensure its validity. Then, the client sends the request message to the optimal node V_3. The request message includes the data abstract, timestamp, and other relevant details.
- **Pre-prepare:** The optimal node V_3 constructs the received request message into a pre-prepare message, and sends it to each trusted node. The pre-prepare message includes the view number, the message serial number, the received request message and other relevant information.
- **Prepare:** The trusted nodes verify the pre-prepare message received from the optimal node. When the validation is successful, the trusted nodes generate the prepare messages and broadcast them to other nodes.
- **Commit:** The optimal node and trusted nodes check whether the received prepare messages are consistent with the pre-prepare messages. If they are consistent, the optimal node generates a new block containing the request message and adds it to the data resource sharing consortium blockchain. The trusted nodes synchronize their ledgers and send commit messages to other consensus nodes.
- **Reply:** All consensus nodes record the received commit messages in their ledgers and send a reply to the client. Once the client receives replies, it indicates that the consensus process is complete, and the requested operation has been agreed upon and executed by the consensus nodes.

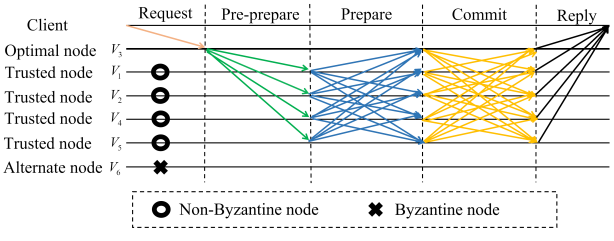


Fig. 5 Consensus process based on PBFT.

4. Simulation Result

Two data resource sharing scenarios for distribution power grid are considered for simulations. One is to set the number of blockchain nodes as 50 and perform 100 consensus processes. The other one is to set the number of blockchain nodes to increase from 20 to 80, and take the average value of 100 iterations as the simulation result. Besides, set proportion of Byzantine node as 20% in both scenarios. The simulation parameters [35], [36] are summarized in Table 2.

The proposed algorithm is compared with two algorithms to verify its performance. The first one is threshold signature-based efficient Byzantine fault-tolerant negotiation algorithm (TS-PBFT) [37], which optimizes consensus node selection by threshold signature techniques to reduce the consensus delay. However, TS-PBFT does not consider influence of consensus delay and node trust degree on the optimization of consensus node selection. The second one is the conventional PBFT-based consensus algorithm (CPBFT), which achieves consensus according to the principle of majority rule. However, CPBFT needs all nodes to participate in the consensus process, which has a large demand on communication resources and computing resources.

4.1 Consensus Delay Analysis

Figure 6 shows the consensus delay versus iteration. Compared with TS-PBFT and CPBFT, the consensus delay of the proposed algorithm decreases by 9.15% and 17.72%, respectively. The proposed algorithm considers the influence of fluctuations of communication and computing resources, and learns to select consensus node with less consensus delay and higher node trust degree according to the upper confidence bound of node performance. TS-PBFT does not consider the influence of consensus delay on optimization of consensus node selection, which leads to an increase in consensus delay. CPBFT requires all blockchain nodes to

Table 2 Simulation parameters.

Parameter	Value	Parameter	Value
N	50	T	100
α	5	β	1.5
γ_h	0.8	γ_r	10
μ	0.01	θ	0.1

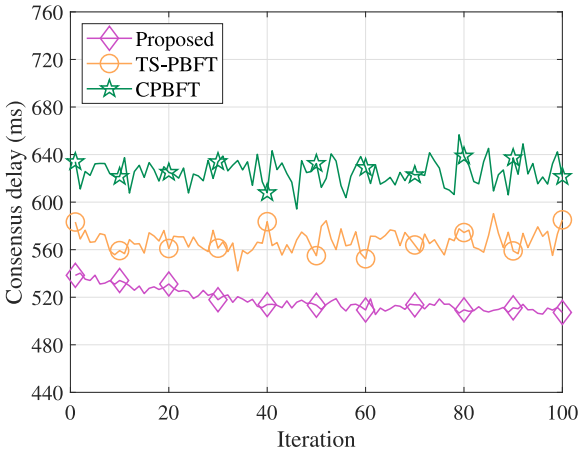


Fig. 6 The consensus delay versus iteration.

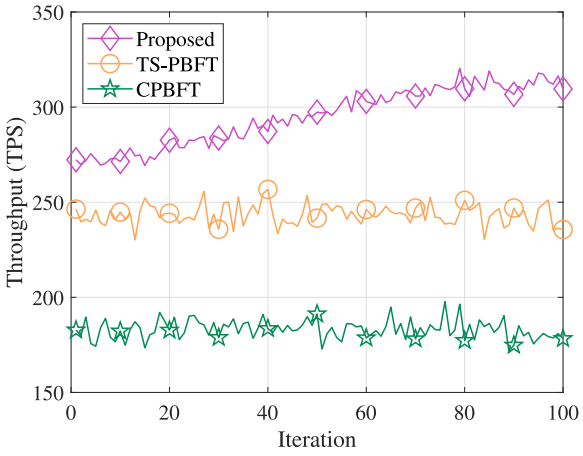


Fig. 8 The throughput versus iteration.

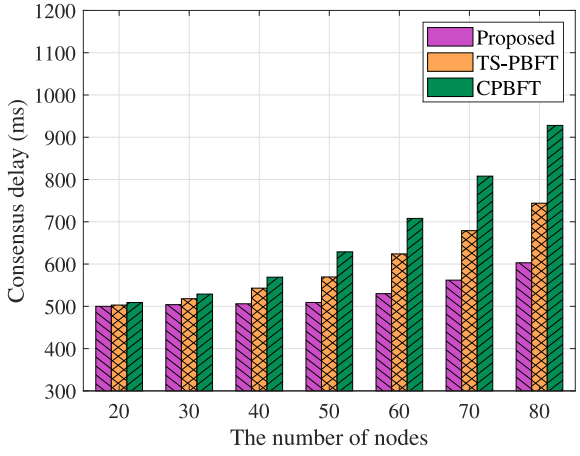


Fig. 7 The consensus delay versus the number of nodes.

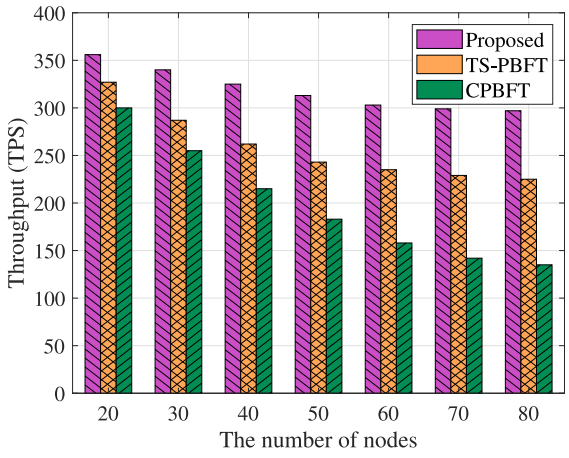


Fig. 9 The throughput versus the number of nodes.

participate in the consensus process, which results in the largest consensus delay.

Figure 7 shows the consensus delay versus the number of nodes. With the number of nodes growing from 20 to 80, compared with TS-PBFT and CPBFT, the growth of consensus delay of the proposed algorithm decreases by 57.26% and 75.42%, respectively. The proposed algorithm divides blockchain nodes into three categories according to the upper confidence bound of node performance, and selects nodes with better performance to participate in the consensus process, which reduces the number of consensus nodes, and has the slowest increase of consensus delay. TS-PBFT cannot guarantee that nodes with better performance are selected to participate in the consensus process. CPBFT require all nodes to participate in the consensus process, which leads to the fastest increase in consensus delay.

4.2 Throughput Analysis

Throughput refers to the number of transactions processed by the algorithm per second, which is measured based on TPS. Throughput is an important index to measure the capability of consensus algorithm to process transactions. Higher throughput indicates that the algorithm is more capable of processing transactions and has higher consensus efficiency.

Figure 8 shows the throughput versus iteration. Compared with TS-PBFT and CPBFT, the throughput of the proposed algorithm increases by 28.81% and 71.04%, respectively. The proposed algorithm learns to select consensus nodes with less consensus delay and higher node trust degree to increase the transactions processed per second, and optimizes the consensus node selection to reduce the number of consensus nodes and communication overhead, which lead to the increase of throughput. TS-PBFT and CPBFT cannot prevent Byzantine nodes from participating in the consensus process, resulting in a large number of consensus nodes, which decreases the throughput due to the significant communication overhead and high consensus delay.

Figure 9 shows the throughput versus the number of nodes. As the number of nodes increases, the communication overhead and consensus delay increase, resulting in fewer transactions processed per second, which lead to the decrease in throughput. When the number of nodes growing from 20 to 80, compared with TS-PBFT and CPBFT,

the throughput degradation of the proposed decreases by 42.16% and 64.24%, respectively. The proposed algorithm considers the historical average performance of node and the fluctuations of communication and computing resources, and classifies nodes according to the upper confidence bound. It ensures non-Byzantine nodes with less consensus delay to participate in the consensus process, and prevents Byzantine nodes from participating in the consensus process, which reduces the number of consensus nodes, and has the smallest throughput degradation. TS-PBFT reduces the number of consensus nodes by threshold signatures techniques to improve throughput, but it cannot accurately identify Byzantine nodes and the throughput cannot be further improved. CPBFT requires all nodes to participate in consensus process, which has a large consensus delay, and the throughput decreases rapidly as the number of nodes increases.

4.3 Security Analysis

The security of data resource sharing can be measured by the proportion of Byzantine node and probability of successful attack. The proportion of Byzantine node refers to the ratio of the number of Byzantine nodes to the number of consensus nodes. The probability of successful attack refers to the probability that consensus cannot be reached due to attack.

There are many kinds of attacks in the blockchain and data resource sharing. This paper takes three typical attacks as examples to analyze the security of the proposed algorithm, which are replay attacks [38], DDoS attacks [39], and sybil attacks [40].

- **Replay attacks:** Replay attack occurs when an attacker maliciously copies or replays valid transactions on different networks or at different times. Replay attack is used to undermine the correctness of verification.
- **DDoS attacks:** DDoS attack combines multiple computers as an attack platform to launch attacks on blockchain nodes, which can generate large amounts of abnormal traffic and interfere with the normal transaction process.
- **Sybil attacks:** Sybil attack disturbs the consensus process by disguising one attacker as multiple fake blockchain nodes with different forged identities, which will mislead the selection of consensus nodes and prevent normal nodes from participating in the consensus process.

Figure 10 shows the proportion of Byzantine node versus iteration. With the increasing of iteration, the proportion of Byzantine node of the proposed algorithm gradually decreases and approaches 0 around the 70-th iteration. The proposed algorithm learns and identifies Byzantine nodes based on upper confidence bound. If a Byzantine node is selected as a consensus node, the blockchain will degrade its node performance according to the node trust degree model after consensus is completed. In the next iteration, the node will not be selected as a consensus node due to the lower upper confidence bound. Therefore, Byzantine nodes will be

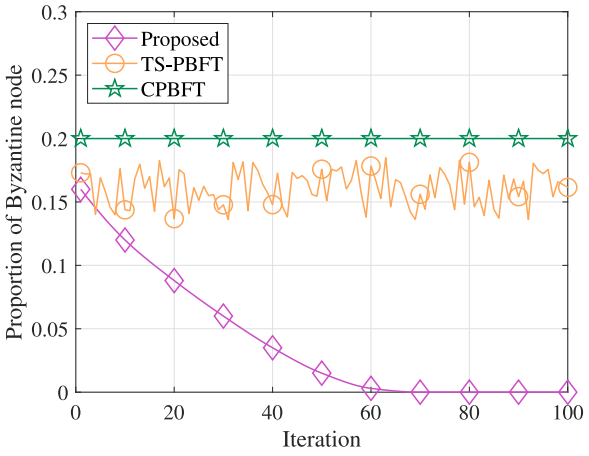


Fig. 10 The proportion of Byzantine node versus iteration.

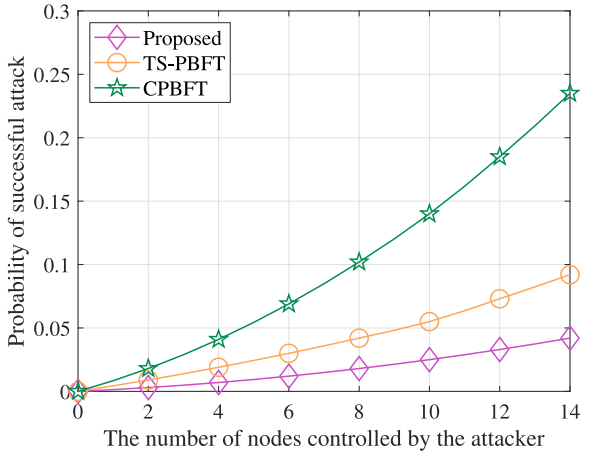


Fig. 11 The probability of successful attack versus the number of nodes controlled by the attacker.

gradually excluded from the consensus process. The proportion of Byzantine node of TS-PBFT fluctuates between 0.13 and 0.19, because it does not consider the influence of node trust degree on consensus node selection optimization, and cannot accurately identify Byzantine nodes. The proportion of Byzantine node of CPBFT is kept constant at 0.2, which is due to the fact that all the nodes participate in consensus process, and the number of consensus nodes and malicious nodes remain constant.

Figure 11 shows the probability of successful attack versus the number of nodes controlled by the attacker. When the number of nodes controlled by the attacker is 14, compared with TS-PBFT and CPBFT, the proposed algorithm reduces the probability of successful attack by 54.35% and 82.12%, respectively. The proposed algorithm can accurately identify Byzantine nodes caused by multiple attack methods and prevent them from participating in the consensus process. TS-PBFT cannot accurately identify Byzantine nodes because it does not consider the impact of node trust degree on the optimization of consensus node selection, and can only prevent some Byzantine nodes from participating in the

consensus process. CPBFT requires all nodes to participate in the consensus process, and cannot reduce the influence of the attacked node on the consensus process.

5. Conclusion

In this paper, we aim to solve the problem of low security and high consensus delay for data resource sharing in distribution power grid. A blockchain-based data resource sharing model was constructed to realize secure data resource storage by combining the consortium blockchain and IPFS. An Improved PBFT consensus algorithm based on intelligent consensus node selection was proposed to minimize consensus delay while improving the security of data resource sharing by optimizing the consensus node selection based on upper confidence bound. The simulation results show that compared with TS-PBFT and CPBFT, the proposed algorithm respectively decreases the consensus delay by 9.15% and 17.72%, and respectively reduces the probability of successful attack by 54.35% and 82.12%. In the future, we will consider more detailed node classification standards and data storage processes to ensure the security of the data resource sharing for distribution power grid.

Acknowledgments

This work was supported by the Science and Technology Project of State Grid Shandong Electric Power Company (Key Technical Research and Application of Real-time Data Sharing Capability Construction for Large-scale Power Gird Nodes, 520608230007).

References

[1] M. Tariq, M. Ali, F. Naeem, and H.V. Poor, "Vulnerability assessment of 6G-enabled smart grid cyber–physical systems," IEEE Internet Things J., vol.8, no.7, pp.5468–5475, 2021.
[2] C. Wang, M. Pang, L. Xu, L. Zhao, Y. Yao, and W. Wang, "Time synchronization and signal detection in non-orthogonal unicast and broadcast networks," IEEE Trans. Broadcast., vol.69, no.2, pp.635–646, 2023.
[3] S. Zhang, H. Liao, Z. Zhou, Y. Wang, H. Zhang, X. Wang, S. Mumtaz, and M. Guizani, "Federated deep actor-critic-based task offloading in air-ground electricity IoT," 2021 IEEE Global Communications Conference (GLOBECOM), pp.1–6, 2021.
[4] M. Tariq, M. Adnan, G. Srivastava, and H.V. Poor, "Instability detection and prevention in smart grids under asymmetric faults," IEEE Trans. Ind. Appl., vol.56, no.4, pp.4510–4520, 2020.
[5] S. Zhao, F. Li, H. Li, R. Lu, S. Ren, H. Bao, J.-H. Lin, and S. Han, "Smart and practical privacy-preserving data aggregation for fog-based smart grids," IEEE Trans Inf. Forensics Security, vol.16, no.1, pp.521–536, Aug. 2021.
[6] S. Zhang, Z. Wang, Z. Zhou, Y. Wang, H. Zhang, G. Zhang, H. Ding, S. Mumtaz, and M. Guizani, "Blockchain and federated deep reinforcement learning based secure cloud-edge-end collaboration in power IoT," IEEE Wireless Commun., vol.29, no.2, pp.84–91, April 2022.
[7] J. Chang, J. Ni, J. Xiao, X. Dai, and H. Jin, "SynergyChain: A multichain-based data-sharing framework with hierarchical access control," IEEE Internet Things J., vol.9, no.16, pp.14767–14778, Feb. 2022.
[8] J. Byabazaire, G.M. O'Hare, and D.T. Delaney, "End-to-end data quality assessment using trust for data shared IoT deployments," IEEE Sensors J., vol.22, no.20, pp.19995–20009, Sept. 2022.
[9] J. Qian, Z. Cao, X. Dong, J. Shen, Z. Liu, and Y. Ye, "Two secure and efficient lightweight data aggregation schemes for smart grid," IEEE Trans. Smart Grid, vol.12, no.3, pp.2625–2637, Dec. 2021.
[10] H. Liao, Z. Zhou, Z. Wang, C. Pan, Z. Jia, and M. Guizani, "Blockchain and learning-based computation offloading in space-assisted power IoT," 2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp.1–6, Dec. 2021.
[11] Z. Shen, F. Ding, Y. Yao, A. Bhardwaj, Z. Guo, and K. Yu, "A privacy-preserving social computing framework for health management using federated learning," IEEE Trans. Comput. Social Syst., vol.10, no.4, pp.1666–1678, 2023.
[12] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "NPP: A new privacy-aware public auditing scheme for cloud data sharing with group users," IEEE Trans. Big Data, vol.8, no.1, pp.14–24, May 2022.
[13] M. Tariq and H.V. Poor, "Electricity theft detection and localization in grid-tied microgrids," IEEE Trans. Smart Grid, vol.9, no.3, pp.1920–1929, 2018.
[14] Y. Ju, M. Yang, C. Chakraborty, L. liu, Q. Pei, M. Xiao, and K. Yu, "Reliability-security tradeoff analysis in mmWave ad hoc based CPS," ACM Trans. Sen. Netw., vol.20, no.2, pp.1–23, 2024.
[15] Y. Wang, Z. Su, N. Zhang, J. Chen, X. Sun, Z. Ye, and Z. Zhou, "SPDS: A secure and auditable private data sharing scheme for smart grid based on blockchain," IEEE Trans. Ind. Informat., vol.17, no.11, pp.7688–7699, Nov. 2021.
[16] C. Xu, Y. Qu, T.H. Luan, P.W. Eklund, Y. Xiang, and L. Gao, "A lightweight and attack-proof bidirectional blockchain paradigm for internet of things," IEEE Internet Things J., vol.9, no.6, pp.4371–4384, Aug. 2022.
[17] H. Liao, Z. Wang, Z. Zhou, Y. Wang, H. Zhang, S. Mumtaz, and M. Guizani, "Blockchain and semi-distributed learning-based secure and low-latency computation offloading in space-air-ground-integrated power IoT," IEEE J. Sel. Topics Signal Process., vol.16, no.3, pp.381–394, Dec. 2022.
[18] S. Zhang, Z. Wang, Z. Zhou, Y. Wang, H. Zhang, G. Zhang, H. Ding, S. Mumtaz, and M. Guizani, "Blockchain and federated deep reinforcement learning based secure cloud-edge-end collaboration in power IoT," IEEE Wireless Commun., vol.29, no.2, pp.84–91, 2022.
[19] Q. He, Z. Feng, H. Fang, X. Wang, L. Zhao, Y. Yao, and K. Yu, "A blockchain-based scheme for secure data offloading in healthcare with deep reinforcement learning," IEEE/ACM Trans. Netw., vol.32, no.1, pp.65–80, 2024.
[20] G. Yang, K. Lee, K. Lee, Y. Yoo, H. Lee, and C. Yoo, "Resource analysis of blockchain consensus algorithms in hyperledger fabric," IEEE Access, vol.10, no.1, pp.74902–74920, July 2022.
[21] Z. Zhou, Y. Wan, Q. Cui, K. Yu, S. Mumtaz, C.N. Yang, and M. Guizani, "Blockchain-based secure and efficient secret image sharing with outsourcing computation in wireless networks," IEEE Trans. Wireless Commun., vol.23, no.1, pp.423–435, 2024.
[22] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M.A. Imran, "A scalable multi-layer PBFT consensus for blockchain," IEEE Trans. Parallel Distrib. Syst., vol.32, no.5, pp.1146–1160, Dec. 2021.
[23] M. Yasin Kubilay, M. Sabir Kiraz, and H. Ali Mantar, "KORGAN: An efficient PKI architecture based on PBFT through dynamic threshold signatures," The Computer Journal, vol.64, no.1, pp.564–574, Nov. 2019.
[24] X. Zhang, R. Li, and H. Zhao, "A parallel consensus mechanism using PBFT based on DAG-lattice structure in the internet of vehicles," IEEE Internet Things J., vol.10, no.6, pp.5418–5433, Nov. 2023.
[25] H.N. Abishu, A.M. Seid, Y.H. Yacob, T. Ayall, G. Sun, and G. Liu, "Consensus mechanism for blockchain-enabled vehicle-to-vehicle energy trading in the internet of electric vehicles," IEEE Trans. Veh. Technol., vol.71, no.1, pp.946–960, Nov. 2022.

[26] S. Zhang, Y. Zhang, and B. Wang, "Antiquantum privacy protection scheme in advanced metering infrastructure of smart grid based on consortium blockchain and RLWE," IEEE Syst. J., vol.17, no.2, pp.3036–3046, March 2023.
[27] W. Tong, X. Dong, and J. Zheng, "Trust-PBFT: A PeerTrust-based practical Byzantine consensus algorithm," 2019 International Conference on Networking and Network Applications (NaNA), pp.344–349, March 2019.
[28] H. Qushtom, J. Misic, V.B. Misic, and X. Chang, "A two-stage PBFT architecture with trust and reward incentive mechanism," IEEE Internet Things J., vol.10, no.13, pp.11440–11452, Feb. 2023.
[29] Y. Jiang, Y. Le, J. Wang, and X. You, "GaS-PBFT: A game-based node selection consensus mechanism for internet of things," 2022 14th International Conference on Wireless Communications and Signal Processing (WCSP), pp.17–21, Feb. 2022.
[30] Z. Pang, Y. Yao, Q. Li, X. Zhang, and J. Zhang, "Electronic health records sharing model based on blockchain with checkable state PBFT consensus algorithm," IEEE Access, vol.10, no.1, pp.87803–87815, July 2022.
[31] Z. Ma, L. Wang, and W. Zhao, "Blockchain-driven trusted data sharing with privacy protection in IoT sensor network," IEEE Sensors J., vol.21, no.22, pp.25472–25479, Dec. 2021.
[32] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-PBFT: An eigentrust-based practical Byzantine fault tolerance consensus algorithm," China Commun., vol.16, no.12, pp.111–123, Dec. 2019.
[33] J. Xiang, J. Zhao, W. Zhou, D. Wang, Q. Ai, S. Yin, and L. Qi, "Consensus mechanism of virtual power plant transaction blockchain based on credit value," 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST), pp.527–533, March 2022.
[34] S. Ma, S. Wang, and W.T. Tsai, "Delay analysis of consensus communication for blockchain-based applications using network calculus," IEEE Wireless Commun. Lett., vol.11, no.9, pp.1825–1829, June 2022.
[35] H. Xu, Y. Fan, W. Li, and L. Zhang, "Wireless distributed consensus for connected autonomous systems," IEEE Internet Things J., vol.10, no.9, pp.7786–7799, Dec. 2023.
[36] C. Wang, M. Pang, G. Cui, X. Chang, F. Jiang, Y. Yao, and W. Wang, "Joint waveform design and detection in symbiotic ambient backscatter NOMA systems," IEEE Internet Things J., vol.10, no.22, pp.19507–19517, 2023.
[37] W. Jiang, L. Chen, Y. Wang, and S. Qian, "An efficient Byzantine fault-tolerant consensus mechanism based on threshold signature," 2020 International Conference on Internet of Things and Intelligent Applications (ITIA), pp.1–5, Feb. 2020.
[38] P. Ramanan, D. Li, and N. Gebraeel, "Blockchain-based decentralized replay attack detection for large-scale power systems," IEEE Trans. Syst., Man, Cybern., Syst., vol.52, no.8, pp.4727–4739, Aug. 2022.
[39] R. Chaganti, R.V. Boppana, V. Ravi, K. Munir, M. Almutairi, F. Rustam, E. Lee, and I. Ashraf, "A comprehensive review of denial of service attacks in blockchain ecosystem and open challenges," IEEE Access, vol.10, no.1, pp.96538–96555, Sept. 2022.
[40] A. Hafid, A.S. Hafid, and M. Samih, "A tractable probabilistic approach to analyze Sybil attacks in sharding-based blockchain protocols," IEEE Trans. Emerg. Topics Comput., vol.11, no.1, pp.126–136, June 2023.

Zhimin Shao received the B.S. degree in Hunan University. He is working in State Grid Shandong Electric Power Company. He has participated in several national-level projects and provincial and ministerial level projects. He works in the field of power internet of things and distribution power grid energy dispatching.

Chunxiu Liu received the B.S. degree in Institute of High Energy Physics, Chinese Academy of Sciences. She is working in State Grid Shandong Dezhou Power Supply Company. She is engaged in the professional management of power system automation.

Cong Wang is working in State Grid Shandong Dezhou Power Supply Company. His works mainly focus on smart grids, 6G communication networks, and PLC technology.

Longtan Li is working in State Grid Shandong Pingyuan Power Supply Company. His works mainly focus on wireless networks, smart girds, and distributed data sharing.

Yimin Liu is working in State Grid Shandong Dezhou Power Supply Company. Her works focus on sensing technologies to realize the state of distribution network equipment and distribution network power quality management.

Zaiyan Zhou is working in State Grid Shandong Dezhou Power Supply Company. His works mainly focus on the highly reliable power supply and cyber-physics systems.