TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/18 pp69–81
DOI: 10.26599/TST.2021.9010074
Volume 28, Number 1, February 2023

A Trust-Based Hierarchical Consensus Mechanism for Consortium Blockchain in Smart Grid

Xingguo Jiang, Aidong Sun*, Yan Sun, Hong Luo, and Mohsen Guizani

Abstract: As the smart grid develops rapidly, abundant connected devices offer various trading data. This raises higher requirements for secure and effective data storage. Traditional centralized data management does not meet the above requirements. Currently, smart grid with conventional consortium blockchain can solve the above issues. However, in the face of a large number of nodes, existing consensus algorithms often perform poorly in terms of efficiency and throughput. In this paper, we propose a trust-based hierarchical consensus mechanism (THCM) to solve this problem. Firstly, we design a hierarchical mechanism to improve the efficiency and throughput. Then, intra-layer nodes use an improved Raft consensus algorithm and inter-layer nodes use the Byzantine Fault Tolerance algorithm. Thirdly, we propose a trust evaluation method to improve the election process of Raft. Finally, we implement a prototype system to evaluate the performance of THCM. The results demonstrate that the consensus efficiency is improved by 19.8%, the throughput is improved by 12.34% , and the storage is reduced by 37.9%.

Key words: consortium blockchain; consensus algorithm; trust evaluation method; smart grid; Internet of Things (IoT)

1 Introduction

Smart grid, as one kind of the industrial Internet of Things (IIOT), can effectively improve the reliability, flexibility, and quality of power grid[1, 2]. Nowadays, there are a large number of matching power generation equipment, substation equipment, and electrical equipment[3–5]. These devices usually produce abundant transaction data onto the process of generating and transmitting power, such as power generation, device usage time, and power usage[6, 7]. Figure 1 illustrates the common hierarchical structure of smart grid[8]. It mainly includes a substation, power plant, and smart meter at different network levels. The main duty of the smart grid is to transport energy in a stable form. Therefore, it is important to ensure the security, stability, and resilience of transaction data between different devices in power grid[9].

To manage power grid data, there are often two solutions: centralized and decentralized methods[10]. For example, IPSO[11], COTCA[12], and DETA[13] are centralized approaches that utilize independent controllers to solve the problem of data management or optimize the network structure. Because of the large scale of smart grid, these solutions may cause poor network performance and redundant data records[14]. Therefore, it is more reasonable to adopt the distributed management mode in smart grid. At the same time, as the blockchain can guarantee trust, security, elasticity, transparency, and scalability for the preserved data, it attracts great attention in smart grid[9, 15, 16].

According to network scope, the blockchain can be divided into a public chain, private chain, and consortium chain[17]. The public blockchain based on proof-of-work (POW) mechanism generally has the problems of slow transaction speed, low efficiency, and small account size[18]. As a semi-open blockchain system, the consortium chain has advantages in

• Xingguo Jiang, Yan Sun, and Hong Luo are with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: jxgaugusto@gmail.com; sunyan@bupt.edu.cn; luoh@bupt.edu.cn.
• Aidong Sun is with Institute of Food safety and Nutrition, Jiangsu Academy of Agricultural Sciences, Nanjing 210000, China. E-mail: sunad2002@163.com.
• Mohsen Guizani is with the Computer Science and Engineering Department, Qatar University, Doha 2713, Qatar. E-mail: mguizani@ieee.org.
∗ To whom correspondence should be addressed.
Manuscript received: 2021-10-01; accepted: 2021-10-13

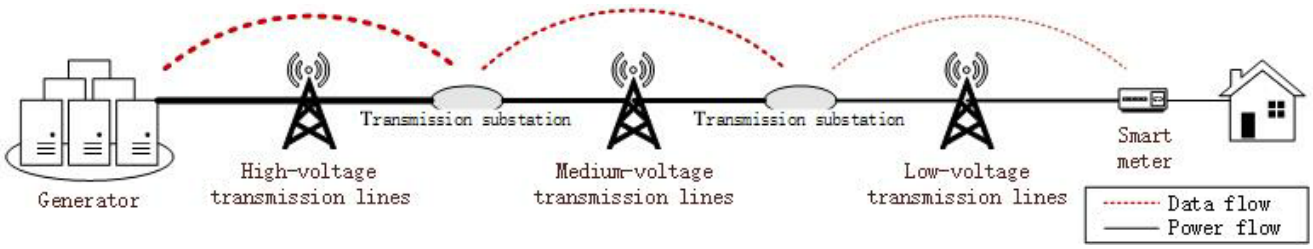


Fig. 1 Common hierarchical structure of smart grid.

identity authentication, consensus efficiency, and design flexibility. Therefore, the consortium chain is more suitable for storing data in smart grid. As the core of blockchain, the consensus algorithm is to make all nodes reach a consensus to complete the data storage, and thus has important influences on service performance. For consortium blockchain, the current consensus algorithm is mainly divided into two categories[19]: one is the Byzantine fault tolerance (BFT) algorithm, such as PBFT[20]; the other is Fail-Stop Failure algorithm, such as Raft[21]. Generally, Raft is more efficient than PBFT, but Raft cannot tolerate Byzantine error. Additionally, the network splitting in Raft may seriously reduce network efficiency. The existing related studies, such as T-PBFT[22] and Kraft[19], mainly focus on algorithm efficiency by grouping and re-establishing the node relationship. However, they may incur low efficiency and data redundancy with an increasing number of nodes.

In order to solve the above problems, we propose a trust-based hierarchical consensus mechanism (THCM). The transactions can be quickly agreed within each layer and some important messages are agreed between layers. Therefore, the first sub-problem is the intra-layer fast consensus. We improve the Raft algorithm to reduce the frequency of split and election. Specifically, the candidate nodes are first selected based on the trust ranking to conveniently complete the election process. Moreover, in order to achieve efficient data storage, in the transaction synchronization process[23], we select the nodes with higher trust to synchronize the data, while the other nodes only synchronize the data hash. The second sub-problem is the inter-layer consensus. When some important transactions need to be distributed among different layers, we use PBFT algorithm to prevent malicious nodes from sending data. The last sub-problem is trust evaluation for network nodes. We improve the Eigen trust model[24] to evaluate trust in two dimensions: communication and storage. The communication and storage data are sent to the leader node regularly through P2P network[25], and the trust result is calculated by the improved model. Then, we select a group of trusted nodes by the trust ranking as the leader candidate nodes to speed up the transition process and reduce the network split[26].

In this paper, our contributions are summarized as follows.

(1) We propose a hierarchical lightweight and efficient consensus algorithm to solve the problem of efficient and secure data storage in large-scale smart grid.

(2) By combining the trust evaluation with the Raft algorithm, we successfully reduce the election time, accelerate the consensus process, and decrease redundant data.

(3) We develop an experimental platform based on FISCO BCOS to evaluate THCM. The results show that it can effectively improve consensus efficiency, increase throughput, and reduce network pressure.

We organize the rest of this paper by the following order. We describe the related work of this paper in Section 2. Next, we present the model design and detailed process and implementation method of the core proposed algorithms in Sections 3 and 4, respectively. Moreover, we record the evaluation results and analysis in Section 5. Finally, a conclusion of this paper is drawn in Section 6.

2 Related Work

In this section, we review and summarize the development of smart grid, the consensus algorithm in consortium blockchain, and the application of consortium blockchain technology in smart grid.

Blockchain is a specific distributed shared database, which is proved to have significant advantages, including security, invariance, and decentralization[27]. It allows each transaction to be recorded in a verifiable and permanent manner, which is essential for creating a distributed, transparent, and secure decentralized energy trading environment. Blockchain experiences rapid development from version 1.0 to version 3.0. Although blockchain 1.0 and 2.0 are more closely

related to bitcoin, cryptocurrency, and transfer contract or property, blockchain 3.0 expands its application fields from financial transactions to a wider range of fields, including energy, education, government, health, and so on[28]. Li et al.[29] proposed a point-to-point IIoT energy trading system based on blockchain, which uses credit-based payment strategy to reduce transaction confirmation delay. Aggarwal et al.[30] proposed a blockchain model called EnergyChain, which is used to realize secure energy transactions between smart grid and smart home, involving mining node selection, block creation and verification, and transaction processing. Aitzhan and Svetinovic[31] implemented the concept verification of distributed energy trading system by using blockchain technology, multi signature, and anonymous encrypted message flow, enabling peers to negotiate energy prices anonymously and execute transactions safely. Sikeridis et al.[32] introduced a novel distributed network architecture based on blockchain, and realized the direct secure communication between smart grid relays, to enhance the security of data between the relays in smart grid. Baza et al.[33] proposed a charge coordination mechanism based on blockchain, which uses an anonymous signature to carry the power information requested in the trading, and it adopts smart contracts to schedule priority to achieve effective billing management.

However, most of the above methods take advantage of workload proof, resulting in an inefficient consensus process, redundant storage, long transaction duration, and other issues[9]. The consortium algorithm improves the network efficiency to some extent. The consortium chain is usually based on non-POW algorithms. There are currently BFT algorithms to solve Byzantine problems, such as PBFT[20], and algorithms that cannot solve Byzantine problems, such as Raft[21]. There are the following improvements for these two algorithms: Gao et al.[22] proposed a group of nodes based on trust and combined with the PBFT algorithm, which can further enhance the robustness of the master group through group signature and mutual supervision. Tong et al.[34] proposed a Trust-PBFT which combines PeerTrust P2P trust computing model with PBFT consensus algorithm to enhance the scalability of the network. Wang et al.[19] presented a Raft-like consensus algorithm Kraft, which optimizes the leader election and consensus process of the Raft consensus algorithm through the K-Bucket node relationship established in the Kademlia protocol, and it improves the speed and throughput of the leader election. Li et al.[35] proposed a POV algorithm based on voting proof. The consensus is coordinated nodes controlled by consortium partners, and these nodes will conduct decentralized arbitration through voting. Wang et al.[36] proposed a new consensus agreement, the Byzantine fault tolerance of credit authorization. The scheme is based on a voting reward and punishment scheme and makes a corresponding credit evaluation to reduce the participation of abnormal nodes.

All of the above researches improve the consensus algorithm of consortium chain in different degrees[37], but rarely focus on the unique requirements of smart grid. In the smart grid, there are some researches for power transaction. Kang et al.[38] proposed a peer-to-peer electricity trading model using a consortium chain, which realizes the iterative double bidding mechanism by using an intelligent contract, digital signature technology, and asymmetric encryption technology, and maximizes the pricing revenue from the perspective of access control. Gai et al.[39] presented a method called privacy preserving blockchain enabled transaction (PBT) model, which organizes interference activities by creating noise. In order to prevent attacks based on data mining and ensure accurate incomplete transaction records, the method uses account mapping mechanism to complete node assignment. Zhou et al.[40] designed a consensus algorithm for the selection of private key generator (PKG) in smart grid, which has a reward and punishment mechanism. At the same time, they designed a blockchain based access control scheme using combination cryptosystem. Fan and Zhang[41] proposed a smart grid data aggregation and regulation mechanism based on a consortium chain. Its signcryption algorithm can be applied to multidimensional data collection and multiple receivers in consortium blockchain. Each receiver analyzes multidimensional data and formulates corresponding control strategies for a single user. Power grid operators implement user power regulation through feedback to smart contracts. Bansal et al.[42] proposed an intelligent extensible ledger framework that does not need too much computational complexity. Nodes which are part of the transaction directed acyclic graph (DAG) are selected for verification. By checking the height of any peer relative to the DAG, the final check of the same object can be verified.

In summary, we find that in the current smart grid transactions, the combination scenarios of consortium chain mostly exist in the access control and privacy protection scenarios, and the improved transaction

efficiency is still less. In other conditions, the improvement in consortium chain algorithm is mostly oriented to BFT algorithm, and hierarchical fusion algorithm is also less. According to our investigation, the THCM is a suitable solution, especially in saving blocks, improving efficiency, and increasing throughput. The next section introduces the design of the model.

3 System Architecture

Energy demand is rapidly increasing owing to new technologies such as electric vehicles. In order to meet the increased energy demand, smart grid architecture is evolving from centralization to decentralization[43]. The decentralized network architecture is shown in Fig. 2.

There are several generation nodes in the network, which are connected with the core transmission nodes, and the core transmission nodes constitute the core network. Each core transmission node is connected to a primary distribution network, which is composed of primary distribution nodes. Each primary distribution node is connected with a secondary distribution network, and the secondary network is composed of secondary distribution nodes. The secondary distribution network is reconnected to the power network composed of smart meters.

In each of the above core networks and subnets, the group nodes share the general information, such as control, management, and trading information, by an intra-layer consensus mechanism. Correspondingly, all the nodes in the above whole network form a large group at the same time, which is used to share important inter-layer information such as flow information, major failure information, and adjustment of important transaction data. As these nodes are transmission nodes with a computer room and an intranet system in the power grid[6], each node in the network can deploy the consortium blockchain function to become a blockchain node.

We design four node types: leader, follower, candidate, and key nodes. Leader nodes are responsible for hosting the consensus process and data storage. Key nodes are responsible for participating in the consensus process and storing data at the same time. The follower nodes are responsible for synchronizing the data hash to ensure that the data can not be tampered with. As the alternative nodes of the leader nodes in the voting process, candidate nodes accept votes from other nodes, and become the leader nodes if more votes are obtained.

As illustrated in Fig. 2, the leader node is the red spere, the key node is blue, and the follower node is black. The blue line in the purple circle represents the intra-layer consensus algorithm, and the black line in the purple circle represents the inter-layer consensus algorithm. Based on the level of consensus data, nodes can participate in the election of this layer, and they also can participate in the election between layers. They may become leaders or other two identities.

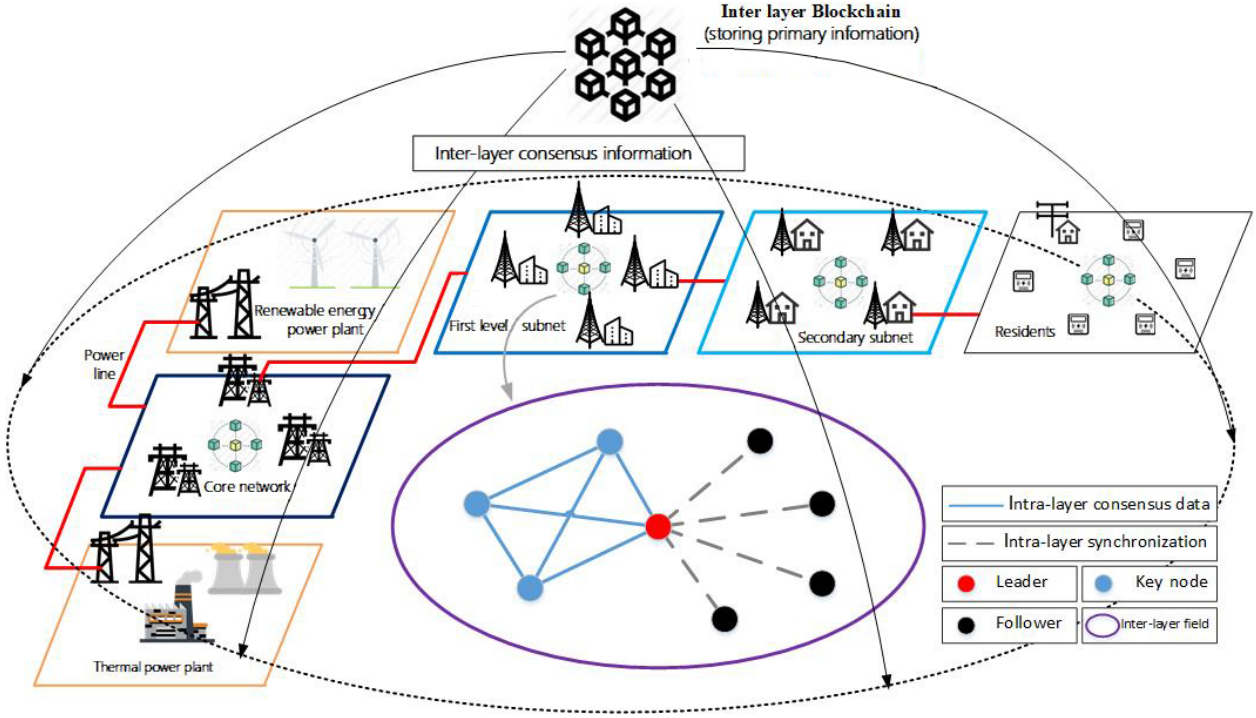


Fig. 2 Hierarchical decentralized network architecture of smart grid based on consortium blockchain.

4 Solution

In this section, the implementation of the THCM is explained in detail, which includes the intra-layer algorithm, inter-layer algorithm, and trust update mechanism.

As mentioned in Section 3, general control and management information only needs to be intra-layer stored; but for power flow or major fault information, the whole network needs to participate in consensus. Thus, the information has different importance levels, and the system adopts different consensus strategies for different levels of data.

Therefore, the blockchain structure of each node is shown in Fig. 3. Each node may have data or a data hash. The black block represents the storage block of secondary data, while the red block represents the storage block of primary data.

4.1 Intra-layer algorithm

This part mainly introduces the improved intra-layer algorithm in the layer. The algorithm is lightweight, which aims to achieve a faster intra-layer consensus process, reduce the election of leader nodes, and make the intra-layer algorithm more efficient, which is divided into two parts: voting and consensus.

4.1.1 Voting

Because of the network delay or leader downtime, the Raft algorithm votes frequently, which greatly wastes bandwidth and causes consensus delay[44]. To avoid this problem, the key node group is introduced to solve this problem. Suppose that N nodes are in this layer, and $m\%$ of nodes need to be selected to form a key node group (here m is 50[22]).

The voting mechanism of the intra-layer algorithm is based on the tenure system and timer mode of Raft algorithm. That is, each message in the system will have two terms of election. One is the term of the current key node group named *Term_all*, and another is the current leader's term named *Term_key*. The two timers correspond to two timeouts named *Term_all_timeout* and *Term_key_timeout*. When a node sends a message, it will start a corresponding timer. Once the timer times out, it will trigger different election processes.

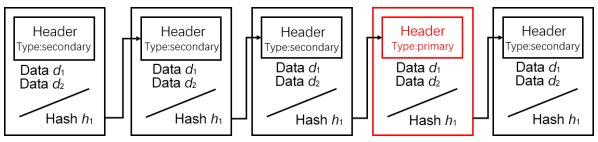


Fig. 3 Blockchain structure of nodes.

The key node group is composed of several nodes with a high degree of trust, which is voted on by all nodes. The function of key nodes is as follows:

- Convenient for quick conversion of leader.
- Convenient for deciding on the storage on node for balanced storage.
- Take the key nodes of this term as the prior condition of the next trust election to make the trust evaluation converge quickly and prevent the emergence of singularity.

In each heartbeat synchronization, the key nodes and the follower nodes start a timer randomly within the corresponding timeout. The *Term_key_timeout* is shorter than the *Term_all_timeout* (the sets of timeout are determined by the experimental data). Therefore, two types of election processes occur: leader election caused by *Term_key_timeout* of key nodes and key node group election caused by *Term_all_timeout* of common nodes.

Election process of the leader. When the key node does not receive the message from the leader within the *Term_key_timeout*, it is considered that the leader is invalid. Then the key node which is time-out firstly initiates a vote to other key nodes and changes into the candidate node. The other key nodes that receive the vote request immediately send the vote to the candidate node.

When the candidate node collects $\lceil \frac{(N\times m+1)}{2} \rceil$ votes, it will automatically become a new leader node, and *Term_key* will be increased by one and then preside over the next term of voting process.

When the node receives the heartbeat message in which the *Term_key* is larger than its own *Term_key*, the node updates the current *Term_key* and takes the source node as its leader, which means the leader node has completed the leader election process.

Every time the leader completes the trust update, it will calculate the new trust result. If the average trust degree of the key node is lower than the highest trust degree of the non key group, the voting message of the key node group is broadcast to start the election process of the key node group.

In this condition, the node has two final states: the leader or key node. The leader is responsible for hosting the consensus process of this layer, while the follower, as the key node, is responsible for data storage. In the intra-layer algorithm, the generation of the leader node depends entirely on the timeout of the random timer, so it has better fairness. Therefore, the algorithm is shown

as Algorithm 1.

Election process of the key node group. Algorithm 2 outlines the election process of node type. When a node becomes a leader node, the node presides over the consensus process. The leader sends heartbeat messages to all nodes. After that, the leader node decides who will be the key nodes. The decision method is as follows: the leader node ranks the trust degree of all the nodes, and sends the corresponding identity information in the heartbeat packet. When other nodes receive heartbeat packets, they will automatically turn into key nodes or follower nodes. At this point, the election process is completed.

4.1.2 Intra-layer consensus process

When the client generates secondary messages, it needs to make consensus on the newly generated data. The process of consensus on the chain follows the order of storage first and then consensus. The complete intra-layer consensus process is given in Algorithm 3. Firstly, the key nodes store the data, and then all nodes write the hash to the consortium chain. The storage of consortium chain adopts the way of database, which is not only convenient to read and write, but also easy to manage and deploy. After all nodes are chained, the current communication status of each node and the remaining capacity of the node are upload to the leader.

Algorithm 1 Leader election

Input: Timer T_1, voting request, *Term_key*
Output: Node type

while not receive term greater than or equal to current *Term_key* in the heartbeat packet **do**
  if T_1 timeout **then**
    $Term\_key = Term\_key + 1$;
    Send leader voting request;
  end if
  if receive leader voting request **then**
    if the first vote in the current *Term_key* **then**
      Send the vote to the candidate node;
    else
      No respond;
    end if
  end if
  if collect $\lceil \frac{(N\times m+1)}{2} \rceil$ votes **then**
    Turn to leader;
    return current type;
  end if
end while
Turn to follower;
return current type;

Algorithm 2 Node type process

Input: Heartbeat packet of leader
Output: All non-leader node types

if a node is leader **then**
  According to the received direct trust of other nodes, the comprehensive trust is calculated iteratively;
  Sort the comprehensive trust;
  Broadcast the node identities;
  return the node types;
else if receive the node type **then**
  Change the node type;
end if

4.2 Inter-layer algorithm

If the primary data are generated by the current client, the process of inter-layer consensus is needed. In this part, PBFT is used, which has two advantages:

- Considering the malicious situation of nodes, PBFT can improve the fault tolerance of the whole system, and improve the data security for important data.
- The generation of important data is random, so PBFT is more compatible with randomness of data generation.

In the process of inter-layer consensus, the leader node of the layer where the data are generated is used as the leader hosting the consensus. Meanwhile, the leader node of other layers is used as the consensus node, and the other nodes are used as the observer node. Therefore, the process of consensus algorithm follows the PBFT

Algorithm 3 Intra-layer consensus process

Input: Node type
Output: Consensus result

if node is leader **then**
  Send the data to the key nodes through heartbeat packet, and send the hash to other node;
  Wait for the feedback until collecting more than $\lceil \frac{(N\times m+1)}{2} \rceil$ packets;
  Leader broadcast the message, then store it into the consortium chain;
  Wait for $\lceil \frac{(N\times m+1)}{2} \rceil$ completed packets and store the communication data and remaining storage capacity shared by each node;
else if node is key node **then**
  Receive heartbeat packet;
  Store the data into the consortium chain;
  Respond to the communication data and remaining storage capacity;
else
  Receive heartbeat packet;
  Store the hash into the consortium chain;
  Respond to the communication data and remaining storage capacity;
end if

algorithm, which has three phases: pre-prepare, prepare, and commit.

4.3 Trust update

According to the previous description, the trust of the node is an important indicator to maintain the normal operation of the system. Therefore, in the running process of the system, it is necessary to update the trust level in the layer at intervals. The update process in the layer is mainly based on the Eigen trust model. However, for this application scenario, the original model has the following problems:

- When initializing the statistics of communication, the system simply adds the communication scores. If the score is negative, the score will be cleared directly, which will weaken the consideration of communication results. For a long-term running system, the method will weaken the influence of communication results on trust.
- In a distributed algorithm scenario, each iteration relies on message feedbacks from other nodes, which increases the communication overhead significantly and affects the trust score at any time. This is not conducive to the stable operation of the system, but also increases the iteration time.
- The trust mechanism only considers communication; it does not consider storage.

Therefore, the following improvements have been made to the algorithm. When node i sends message to node j, i evaluates this communication by δ_{ij} which indicates the times of successful communications, and τ_{ij} which indicates the number of communication failures. At this point, the local trust value s_{ij} can be expressed as the probability of success of the next communication, and the process follows the Beta distribution[45]. Specially, when the node joins in the network, δ_{ij} and τ_{ij} can be initialized to $B(1,1)$, which is the expression of the Beta distribution and then degenerates to a uniform distribution. We think they have an equal probability of success or failure[46]. δ_{ij} and τ_{ij} are the two parameters of the Beta distribution, so direct trust s_{ij} can be expressed as the expectation of the distribution in Eq. (1).

$$s_{ij} = \frac{\delta_{ij}}{\delta_{ij} + \tau_{ij}} \tag{1}$$

Then, s_{ij} is normalized to get the local trust c_{ij}, which is shown as Eq. (2).

$$c_{ij} = \frac{s_{ij}}{\Sigma_j s_{ij}} \tag{2}$$

After getting local trusts matrix C, all nodes get the global trust T by calculating $t_i = (C)^n c_i$ iteratively. The specific procedure is shown as Algorithm 4. Particularly, according to the Markov chain[47], only if the sum of most columns of initial matrix C is equal, such as columns j_1 and j_2 satisfy $\Sigma c_{j_1} = \Sigma c_{j_2}$, the convergence value will be equal, that is, $t_{ij_1} = t_{ij_2}$. This situation occurs randomly and the probability of two nodes having equal trust is very low because of the influence of storage trust and a priori condition.

At the same time, the impact of the node's remaining capacity M on the global trust level needs to be considered. Because we use databases such as MySQL to manage storage, M represents the ratio of the currently used storage of the total storage. In particular, if the storage capacity of one node differs too much from that of other nodes, this ratio is a good one because the consensus data are the same each time, and the consumed storage is the same each time. After getting M, it requires a normalization process: $m_i = \frac{m_i}{\Sigma_i m_i}$, and the parameter λ is introduced to balance the relationship between storage capacity and local trust. At the same time, the trust rank after leader arbitration in the last term is used as P, and the parament a is used to balance the history global trust vector P. The process is expressed as Eq. (3). We use this arbitration result as a reliable prior condition for iterative calculation. Therefore, for each iteration, both M and P conditions need to be combined to make the various nodes converge quickly, and singularity and periodicity are also avoided[25]. The calculation formula is given in Algorithm 5.

$$\vec{t_l}^{(k+1)} = (1-a)[(1-\lambda)C\vec{t_l}^{(k)} + \lambda m_i] + ap_i \tag{3}$$

5 Experiment

This section implements a series of experiments to

Algorithm 4 Update local trust

Input: Number of successful communications with node j: δ_{ij}, number of failed communications with node j: τ_{ij}

Output: Local trust c_{ij}

1: $sum = 0$;
2: Traversal over each node j:
3: $\quad s_{ij} = \frac{\delta_{ij}}{\delta_{ij}+\tau_{ij}}$;
4: $\quad sum = sum + s_{ij}$;
5: Leader broadcast the message that store it into the consortium chain;
6: Traversal over each node j:
7: $\quad c_{ij} = \frac{s_{ij}}{\Sigma_j s_{ij}}$;
8: **return** c_{ij};

Algorithm 5 Update global trust

Input: Local trust matrix C, used storage ratio vector M
Output: Global trust matrix T

$t^0 = p_i$;
for each node i **do**
  while $\delta > \epsilon$ **do**
    $\vec{t}^{k+1} = C\vec{t}^{k}$;
    $\vec{t}^{k+1} = (1-\lambda)\vec{t}^{k+1} + \lambda m_i$;
    $\vec{t}^{k+1} = (1-a)\vec{t}^{k+1} + ap_i$;
    $\delta = ||\vec{t}^{k+1} - \vec{t}^{k}||$;
  end while
end for
return $\vec{t}^{k+1}$;

evaluate the performance of our model. Safety is not the focus of our evaluation because we make improvements based on existing algorithms. Therefore, the safety performance is similar to that of conventional algorithms. We focus on the evaluation of the efficiency, throughput, and storage of the system.

5.1 Simulation setup

5.1.1 System configuration and environment

The configuration principle of the experiment is to simulate the smart grid based on hierarchical architecture, and each layer is made up of server nodes. As the system is hierarchically refined, the number of nodes per layer is not large. Therefore, the range of the number of nodes N within our layer is smaller. The software configuration includes FISCO BCOS (version 2.2.0) running on a PC machine (associative desktop host). The hardware configuration includes Windows 10 operating system, a 2.3 GHz CPU, an Intel Core Version i5, and 16 GB of Memory.

The experimental environment was improved in the FISCO BCOS and written in C++. The number of N is set between 10–60 to simulate scenarios with a certain number of nodes. The three settings involved are related to three parameters, which are the time-out setting for all node exchanges within the layer *Term_all_timeout*, the timeout setting for key nodes within the layer set *Term_key_timeout*, and the selection scale m for the key nodes. We show some experimental results in the remainder.

5.1.2 Compared schemes

Since we optimize the intra-layer consensus part, to demonstrate the effectiveness of the THCM, we compare it with Raft algorithm. We first design experiments to determine two timeouts: *Term_all_timeout* and *Term_key_timeout*. Furthermore, the efficiency of the two algorithms is compared by calculating the difference in the time of completing the same number of elections. The throughput is compared by comparing the number of consensus blocks at the same time between the two algorithms. And storage optimization is compared by observing the difference between occupied storage by producing the same consensus blocks.

5.2 Analysis results

5.2.1 Determination of two timeouts

The voting timeouts *Term_key_timeout* and *Term_all_timeout* mentioned in Section 3 are the key factors that determine the efficiency of the algorithm. When the algorithm runs, the leader presides over the election, *Term_all_timeout* is the threshold time to trigger the transition of non-key nodes. That is, when *Term_all_timeout* is exceeded and no heartbeat is detected by non-key nodes, the leader node is determined to be disconnected and converted to a candidate node to participate in the election. *Term_key_timeout* is the time threshold for triggering a key node voting. To determine two values of better algorithm performance, the following set of experiments is designed.

Determination of *Term_all_timeout*. To determine the optimal value of *Term_all_timeout*, two experiments are designed as follows. The total number of network nodes in the layers is n. The time to complete ten elections with different *Term_all_timeout* and n settings is determined. The way to complete the one election is to keep silent whenever the leader node sends ten heartbeats to trigger the all nodes election.

Because the timeout time of traditional Raft algorithm is set to 1000 ms, according to the convention, the start time of *Term_all_timeout* is set to 1000 ms. To control the variable, *Term_key_timeout* is set to 500 ms. The experimental results are shown in Fig. 4.

The data show that when the number of network nodes and *Term_all_timeout* increase, the time to complete the election basically changes linearly, so 1000 can be

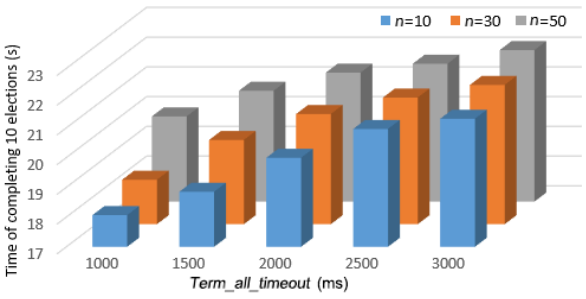


Fig. 4 Comparison of election time of *Term_all*.

selected as the value of *Term_all_timeout*.

The above situation only considers the best network state, that is, the packet loss rate $PE = 0\%$. The real network can not achieve such an ideal situation. We simulate different packet loss rates by preventing the key node and non-key node from sending election packets with a certain probability. The number of nodes in the experimental network is $n = 30$, and the time to complete 10 general elections with different PE values and different *Term_all_timeout* is measured respectively. The experimental results are shown in Fig. 5.

The results show that in each time setting, the growth of packet loss rate PE leads to an increase in the voting time, but the growth rate is roughly the same. With the increase in PE, the timeout of nodes is infrequent, which leads to frequent network collisions. Therefore, the *Term_all_timeout* is set to 1000 ms.

Determination of *Term_key_timeout*. Similar with the determination of *Term_all_timeout*, in order to get *Term_key_timeout*, we design the following two experiments. To avoid collisions of key and non-key nodes with timeout, we set *Term_all_timeout* as 2000 ms, split *Term_key_timeout* into six time intervals from 250 ms to 1500 ms and determine when n nodes complete 10 elections. The result is shown in Fig. 6.

The result shows that with the increase in *Term_key_timeout*, the election time increases gradually and changes linearly. Therefore, we can choose the lower data as the value of *Term_key_timeout*. However, Fig. 6 shows only the data for ideal network conditions. Choose $n = 30$ below to determine the expiration time for PE with different packet loss rates. The experimental results

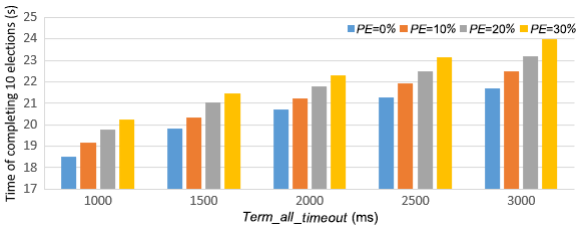


Fig. 5 Comparison of election time of *Term_all* in 30 nodes.

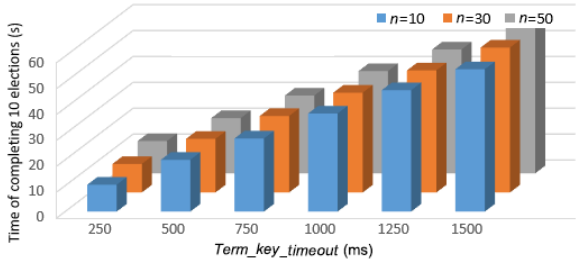


Fig. 6 Comparison of election time of *Term_key*.

are shown in Fig. 7.

The experiment shows that when *Term_key_timeout* is 250 ms, the election time increases significantly with the increase in packet loss rate. This result proves that when the *Term_key_timeout* is too small and the packet loss rate is too high, network collisions will increase significantly, leading to frequent key node election, and the election time will be increased. Therefore, it is a good choice to choose *Term_key_timeout* as 500 ms.

5.2.2 Comparison of efficiency between improved algorithm and traditional algorithm

In order to compare the efficiency of the improved algorithm with that of the traditional algorithm, the following set of experiments is designed to determine the election time of the algorithm for different numbers of nodes and different network packet loss rates. This experiment follows the data of Section 5.2.1, *Term_key_timeout* = 500 ms, *Term_all_timeout* = 1000 ms, our goal is to compare the time of 10 elections with different node numbers n. If the election time is shorter, we think the algorithm has higher efficiency. This experiment simulates different network packet loss rates (0, 10%, 20%, and 30%). The experimental results are shown in Fig. 8.

The experiment shows that the election delay increases with the number of nodes increasing, but the efficiency of the improved algorithm is still better. So calculating the lift rate R_E can compare the efficiency improvements in THCM. Because of the improved algorithm, the total number of nodes in the election is reduced, which improves the speed of re-election. The calculation method of R_E is expressed as Eq. (4).

$$R_E = \frac{Time_{traditional} - Time_{improved}}{Time_{traditional}} \tag{4}$$

Figure 9 compares the improvement in algorithm efficiency at different packet loss rates. As the network packet loss rate increases, the efficiency of the improved algorithm increases gradually. For the traditional algorithms, when the number of nodes is large, the higher packet loss rate will trigger frequent timeouts

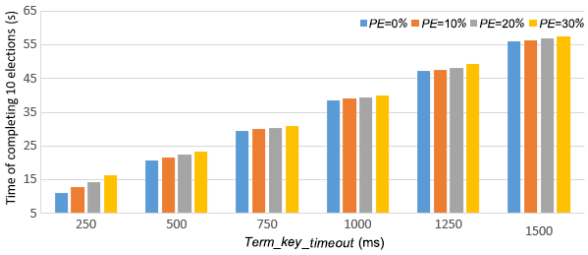


Fig. 7 Comparison of election time of *Term_key* in 30 nodes.

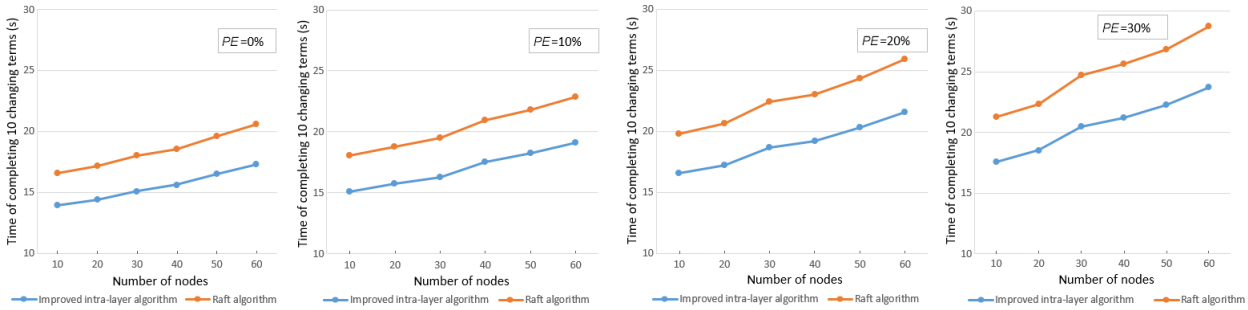


Fig. 8 Comparison of efficiency in different network conditions.

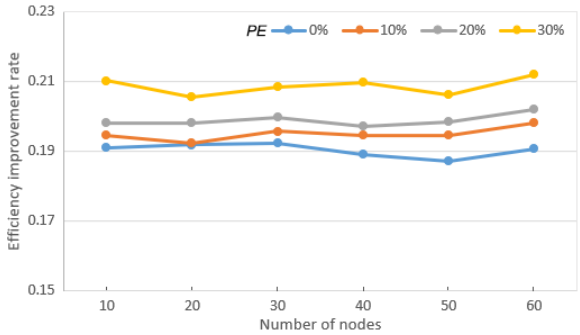


Fig. 9 Comparison of efficiency improvement in different numbers of nodes with four *PE* (0%, 10%, 20%, and 30%).

of election, and there will be some degree of collision, resulting in an increase in the renewal time. For the improved algorithm, nodes with a high degree of trust will participate in the election firstly, to a certain extent, this design avoiding collisions, resulting in the efficiency of the election further improved.

5.2.3 Comparison of throughput between improved algorithm and traditional algorithm

In order to compare the throughput of the improved algorithm with that of the traditional algorithm, the following experiments are designed. Compare the throughput performance of the two algorithms as the number of different nodes increases.

Because the improved algorithm uses key nodes for data storage, non-key nodes only store data hashes, and have higher efficiency. This section conducts a comparative experiment of throughput R_T which is measured by the number of consecutive blocks in 2 min. The consensus packet size here is 2 M, and the experiment uses 30 to 60 nodes for simulation, and counts the number of output blocks of the traditional algorithm and the improved algorithm, respectively. The experimental results are shown in Fig. 10.

The experimental results show that as the number of nodes increases, the number of consensus times of the

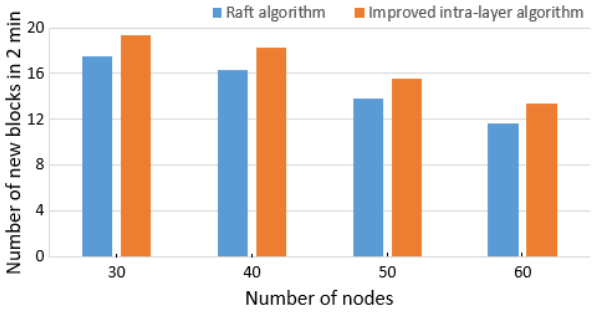


Fig. 10 Throughput comparison in different numbers of nodes between Raft and improved intra-layer algorithm.

improved algorithm is more than that of the traditional algorithm, thus the throughput is improved. Because the improved intra-layer algorithm improves the efficiency of election and uses the mechanism of key nodes to store data and other nodes to store hashes, which reduces the synchronization cost, the block efficiency will also be improved to some extent. Therefore, we calculate R_T which is expressed as Eq. (5) to compare the throughput improvements in the THCM. The result is shown in Fig. 11.

$$R_T = \frac{Throughput_{traditional} - Throughput_{improved}}{Throughput_{traditional}} \times 100\% \tag{5}$$

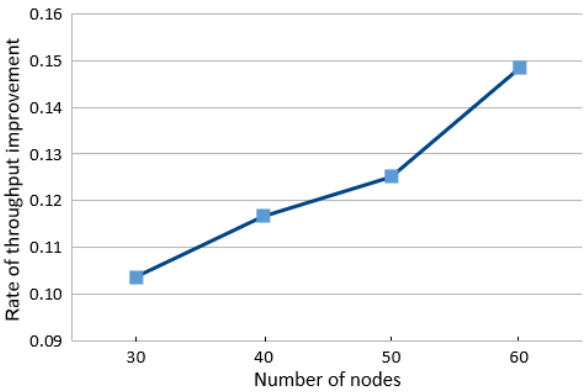


Fig. 11 Rate of throughput comparison in different numbers of nodes.

The results show that the improved algorithm improves throughput to some extent, and the improvement rate increases with the number of nodes. Because the algorithm uses a partial storage mechanism, when there are more nodes, the throughput will be further improved with a shorter block synchronization time than traditional algorithms.

5.2.4 Overall storage comparison

To compare the overall algorithm storage, we take 30 nodes as an example. After running the intra-layer and inter-layer algorithms, the system has reached consensus continuously. We count the total storage changes of all nodes. By changing the total amount of storage, we can compare the advantages of storage. The experimental results are shown in Fig. 12.

The results show that the THCM decreases storage compared to the Raft and PBFT algorithm in different number of new blocks. Because key nodes are used to store data, the mechanism which other nodes store hashes can effectively reduce storage pressure.

6 Conclusion

This paper mainly discusses the storage of transaction data in smart grid and proposes a hierarchical consensus algorithm based on trust evaluation. In the proposed approach, the nodes in the network are layered, and different algorithms are applied between layers and within layers, respectively. The intra-layer nodes use the improved Raft algorithm. We add the key node in the node types, so that the nodes can complete the election rapidly. At the same time, the PBFT algorithm, which can avoid the Byzantine problem, is used in the nodes between layers. The trust evaluation mechanism is introduced in THCM. The communication trust and storage trust are computed through Eigen trust model and Markov chain to select key node groups. Our evaluations provide practical proof for the proposed approach.

In the future, we intend to test the THCM in realistic scenarios and design a more optimized inter-layer consensus method.

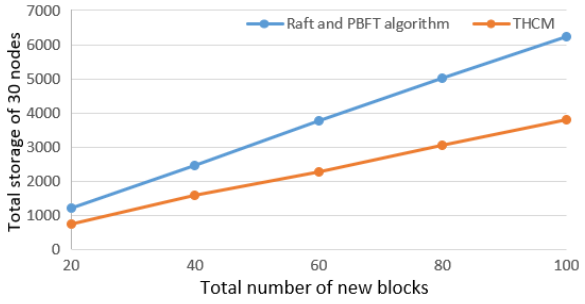


Fig. 12 Total storage volume comparison in different total numbers of new blocks with 30 nodes.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62172051, 61772085, and 61877005) and Jiangsu Agriculture Science and Technology Innovation Fund (No. CX(18)3054).

References

[1] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, PPFA: Privacy preserving fog-enabled aggregation in smart grid, *IEEE Transactions on Industrial Informatics*, vol.14, no. 8, pp. 3733–3744, 2018.

[2] T. Wang, Y. Lu, J. Wang, H. N. Dai, X. Zheng, and W. Jia, EIHDP: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for IoT systems, *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1285–1298, 2021.

[3] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, Blockchain and deep reinforcement learning empowered intelligent 5G beyond, *IEEE Network*, vol. 33, no. 3, pp. 10–17, 2019.

[4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, Mobile edge computing: A survey, *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[5] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, A survey on mobile edge networks: Convergence of computing, caching and communications, *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[6] B. Panajotovic, M. Jankovic, and B. Odadzic, Ict and smart grid, in *Proc. 10^{th} International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services* (*TELSIKS*), Nis, Serbia, 2011, pp. 118–121.

[7] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, All one needs to know about fog computing and related edge computing paradigms: A complete survey, *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[8] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks, *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7992–8004, 2019.

[9] A. S. Musleh, G. Yao, and S. M. Muyeen, Blockchain applications in smart grid–review and frameworks, *IEEE Access*, vol. 7, pp. 86746–86757, 2019.

[10] M. Akhtaruzzaman, M. K. Hasan, S. R. Kabir, S. N. H. S. Abdullah, M. J. Sadeq, and E. Hossain, Hsic bottleneck based distributed deep learning model for load forecasting in smart grid with a comprehensive survey, *IEEE Access*, vol. 8, pp. 222977–223008, 2020.

[11] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, An intelligent auction scheme for smart grid market using a hybrid immune algorithm, *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4603–4612, 2011.

[12] J. Matamoros, D. Gregoratti, and M. Dohler, Microgrids

energy trading in islanding mode, in *Proc. IEEE 3rd International Conference on Smart Grid Communications* (*SmartGridComm*), Tainan, China, 2012, pp. 49–54.

[13] S. Chen, N. B. Shroff, and P. Sinha, Energy trading in the smart grid: From end-user's perspective, in *Proc. 2013 Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2013, pp. 327–331.

[14] M. Pilz and L. Al-Fagih, Recent advances in local energy trading in the smart grid based on game-theoretic approaches, *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1363–1371, 2019.

[15] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, Smart grid metering networks: A survey on security, privacy and open research issues, *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.

[16] P. Ghosh, S. Eisele, A. Dubey, M. Metelko, I. Madari, P. Volgyesi, and G. Karsai, Designing a decentralized fault-tolerant software framework for smart grids and its applications, *Journal of Systems Architecture*, vol. 109, p. 101759, 2020.

[17] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in *Proc. 2017 IEEE International Congress on Big Data* (*BigData Congress*), Honolulu, HI, USA , 2017, pp. 557–564.

[18] S. Xu, X. Chen, and Y. He, Evchain: An anonymous blockchain-based system for charging-connected electric vehicles, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 845–856, 2021.

[19] R. Wang, L. Zhang, Q. Xu, and H. Zhou, K-bucket based raft-like consensus algorithm for permissioned blockchain, in *Proc. 2019 IEEE 25th International Conference on Parallel and Distributed Systems* (*ICPADS*), Tianjin, China, 2019, pp. 996–999.

[20] M. Castro and B. Liskov, Practical Byzantine fault tolerance and proactive recovery, *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[21] L. Lamport, Paxos made simple, *ACM Sigact News*, vol. 32, no. 4, pp. 51–58, 2001.

[22] S. Gao, T. Yu, J. Zhu, and W. Cai, T-PBFT: An eigentrust-based practical byzantine fault tolerance consensus algorithm, *China Communications*, vol. 16, no. 12, pp. 111–123, 2019.

[23] FISCO BCOS, https://fisco-bcos-documentation.readthedocs.io/zh_CN/ latest/index.html, 2021.

[24] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, The eigentrust algorithm for reputation management in P2P networks, in *Proc. the 12th International Conference on World Wide Web* (*WWW2003*), Budapest, Hungary, 2003, pp. 640–651.

[25] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. Yang, and F. Zhou, Access control and authorization in smart homes: A survey, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 906–917, 2021.

[26] T. Wang, Y. Liu, X. Zheng, H. N. Dai, W. Jia, and M. Xie, Edge-based communication optimization for distributed federated learning, *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2021.3083263.

[27] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, Blockchain: A distributed solution to automotive security and privacy, *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.

[28] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, To blockchain or not to blockchain: That is the question, *IT Professional*, vol. 20, no. 2, pp. 62–74, 2018.

[29] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, Consortium blockchain for secure energy trading in industrial internet of things, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, 2018.

[30] S. Aggarwal, R. Chaudhary, G. S. Aujla, A. Jindal, A. Dua, and N. Kumar, Energychain: Enabling energy trading for smart homes using blockchains in smart grid ecosystem, in *Proc. 1st ACM MobiHoc Workshop on Networking and Cybersecurity for Smart Cities*, Los Angeles, CA, USA, 2018, pp. 1–6.

[31] N. Z. Aitzhan and D. Svetinovic, Security and privacy in decentralized energy trading through multisignatures, blockchain and anonymous messaging streams, *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 840–852, 2018.

[32] D. Sikeridis, A. Bidram, M. Devetsikiotis, and M. J. Reno, A blockchain-based mechanism for secure data exchange in smart grid protection systems, in *Proc. 2020 IEEE 17th Annual Consumer Communications Networking Conference* (*CCNC*), Las Vegas, NV, USA, 2020, pp. 1–6.

[33] M. Baza, M. Nabil, M. Ismail, M. Mahmoud, E. Serpedin, and M. A. Rahman, Blockchain-based charging coordination mechanism for smart grid energy storage units, in *Proc. 2019 IEEE International Conference on Blockchain* (*Blockchain*), Atlanta, GA, USA, 2019, pp. 504–509.

[34] W. Tong, X. Dong, and J. Zheng, Trust-PBFT: A peertrust-based practical byzantine consensus algorithm, in *Proc. 2019 International Conference on Networking and Network Applications* (*NaNA*), Daegu, Republic of Korea, 2019, pp. 344–349.

[35] K. Li, H. Li, H. Hou, K. Li, and Y. Chen, Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain, presented at 2017 IEEE 19th International Conference on High Performance Computing and Communications, Bangkok, Thailand, 2017.

[36] Y. Wang, S. Cai, C. Lin, Z. Chen, T. Wang, Z. Gao, and C. Zhou, Study of blockchains's consensus mechanism based on credit, *IEEE Access*, vol. 7, pp. 10224–10231, 2019.

[37] J. Y. Ren, J. Z. Li, H. X. Liu, and T. Qin, Task offloading strategy with emergency handling and blockchain security in sdn-empowered and fog-assisted healthcare IoT, *Tsinghua Science and Technology*, doi: 10.26599/TST.2021.9010046.

[38] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains, *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[39] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, Privacy-preserving energy trading using consortium blockchain in smart grid, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[40] Y. Zhou, Y. Guan, Z. Zhang, and F. Li, A blockchainbased access control scheme for smart grids, in *Proc. 2019*

International Conference on Networking and Network Applications (*NaNA*), Daegu, Republic of Korea, 2019, pp. 368–373.

[41] M. Fan and X. Zhang, Consortium blockchain based data aggregation and regulation mechanism for smart grid, *IEEE Access*, vol. 7, pp. 35929–35940, 2019.

[42] G. Bansal, A. Dua, G. S. Aujla, M. Singh, and N. Kumar, Smartchain: A smart and scalable blockchain consortium for smart grid systems, in *Proc. 2019 IEEE International Conference on Communications Workshops* (*ICC Workshops*), Shanghai, China, 2019, pp. 1–6.

[43] R. Bayindir, I. Colak, G. Fulli, and K. Demirtas, Smart grid technologies and applications, *Renewable & Sustainable Energy Reviews*, vol. 66, pp. 499–516, 2016.

[44] H. Xu, L. Zhang, Y. Liu, and B. Cao, Raft based wireless blockchain networks in the presence of malicious jamming, *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 817–821, 2020.

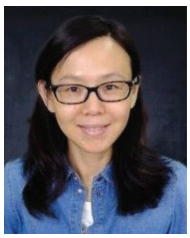
[45] X. S. Zhao and Y. C. Cai, Research of weighting method based on beta distribution, in *Proc. 2015 7^{th} International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, China, 2015, pp. 50–52.

[46] N. Y. Ermolova and O. Tirkkonen, Using beta distributions for modeling distances in random finite networks, *IEEE Communications Letters*, vol. 20, no. 2, pp. 308–311, 2016.

[47] Y. Ephraim and N. Merhav, Hidden markov processes, *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518–1569, 2002.

Xingguo Jiang received the BS degree from Beijing University of Posts and Telecommunications in 2019. He is currently pursuing the PhD degree in Beijing University of Posts and Telecommunications, China. His main research interests include data service and Internet of Things.

Aidong Sun is an associate professor in Jiangsu Academy of Agricultural Sciences. She received the master degree from South China University of Technology in 2005. Her research interests include agri-product traceability system for quality and safety based on Internet of Things.



Yan Sun received the BS degree in information engineering from Beijing Jiaotong University, the MS and PhD degrees in computer applications from Beijing University of Posts and Telecommunications, China, in 1992, 1999, and 2007, respectively. She is a professor with the School of Computer Science, Beijing University of Posts and Telecommunications, China. She is also a research member of Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia. Her research interests include big data, Internet of Things, and software defined networks.

Hong Luo received the BS, MS, and PhD degrees in computer science from Beijing University of Posts and Telecommunications, China, in 1990, 1993, and 2007, respectively. She is a professor of the School of Computer Science, Beijing University of Posts and Telecommunications, China. She is also a research member of Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia. Her research interests include big data, Internet of Things, data service, and communication software.

Mohsen Guizani received the BS (with distinction) and MS degrees in electrical engineering, the MS and PhD degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a professor with the Computer Science and Engineering Department, Qatar University. Previously, he served as the associate vice president of graduate studies, Qatar University, the chair of the Computer Science Department, Western Michigan University, and the chair of the Computer Science Department, University of West Florida. He also served in academic positions at the University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the editor-in-chief of the *IEEE Network Magazine*, serves on the editorial boards of several international technical journals and the founder and editor-in-chief of *Wireless Communications and Mobile Computing Journal* (*Wiley*). He is the author of nine books and more than 500 publications in refereed journals and conferences. He served as the guest editor of a number of special issues in IEEE journals and magazines. He also served as a member, chair, and general chair of a number of international conferences. He received three teaching awards and four research awards throughput his career. He received the 2017 IEEE Communications Society Recognition Award for his contribution to outstanding research in Wireless Communications. He was the chair of the IEEE Communications Society Wireless Technical Committee and the chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005. He is a fellow of IEEE and a senior member of ACM.