Research Article

A Blockchain-Based Microgrid Data Disaster Backup Scheme in Edge Computing

Lihua Zhang ¹, Yu Cao ², Ganzhe Zhang ², Yang Huang ¹ and Chen Zheng ²

¹School of Software, East China Jiaotong University, Nanchang, China

²School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang, China

Correspondence should be addressed to Yu Cao; cylzy2014@163.com

Received 28 May 2021; Revised 7 July 2021; Accepted 24 July 2021; Published 3 August 2021

Academic Editor: Chien Ming Chen

Copyright © 2021 Lihua Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In view of the problems of low security, poor reliability, inability to backup automatically, and overreliance on the third party in traditional microgrid data disaster backup schemes based on cloud backup, the edge computing is used to preprocess power big data, and a microgrid data disaster backup scheme based on blockchain in edge computing environment is proposed in this paper. First, the honey encryption (HE) technology and advanced encryption standard (AES) are combined to propose a new encryption algorithm HE-AES, which is used to encrypt the preprocessed data. Second, the Kademlia algorithm is embedded in the edge server to realize the distributed storage and automatic recovery of microgrid data. Finally, the traditional proof of authority (PoA) consensus mechanism is improved partially, and the improved PoA is used to make each node reach consensus and pack blocks on the chain. The scheme can not only realize the data disaster backup automatically but also has high efficiency of data processing, which can provide a new idea for improving the current data disaster backup schemes.

1. Introduction

It is an important development direction of the world energy technology to promote the proportion of distributed energy system in the energy system and to improve the permeability of new energy with the maturation and perfection of distributed clean energy resources related technology equipment [1]. With the development of the microgrid in recent years, a large number of power data was produced. These power data have a great scientific research value and are of great significance for the optimization of microgrid. However, in the face of natural disasters or disasters caused by improper human operation, it will cause data loss and affect the safe and stable operation of microgrid [2]. Therefore, how to guarantee the safe and reliable disaster backup of microgrid data is difficult.

A large number of temporary data can be processed by edge computing at the edge, and its server is closer to the data source, which can become a feasible choice to process the big data of microgrid [3]. However, the massive data are

stored in the local edge intelligent terminal with the introduction of edge computing, which will bring serious security risks [4].

Blockchain technology has been one of the research hotspots of experts and scholars in recent years due to its characteristics of decentralization, anonymity, data transparency, and traceability [5]. However, the further development of blockchain is restricted because of its limited storage capacity [6].

Therefore, combining edge computing with blockchain, the edge computing is used to process big data of microgrid and reduce data scale, and the blockchain technology is used to ensure the security of edge computing equipment. A blockchain-based microgrid data disaster backup scheme is proposed in edge computing environment, which can solve the problem that data loss is difficult to recover and realize distributed data storage and automatic disaster backup. Compared with other traditional disaster backup schemes, the characteristics of this scheme are as follows.

- (1) A new HE-AES algorithm is used to encrypt data, which can not only enhance the security of data but also resist violent attacks [7].
- (2) The Kademlia algorithm is used to realize distributed storage and disaster backup of data automatically [8], which can greatly improve the routing query speed compared with other distributed technologies [9].
- (3) The PoA is adopted on the blockchain to enable each node to reach consensus and pack the blocks to improve transaction efficiency [10].
- (4) In order to solve the problem that the node information of traditional PoA is easy to be disclosed after multiple rounds of consensus and cannot adapt to Byzantine fault tolerance (BFT) [11], the following three improvements are made to PoA:
 - (a) The verifiable random function (VRF) is used to introduce committee-endorsing mechanism to improve the selection process of authorized nodes [12],
 - (b) The PageRank algorithm and Pareto distribution are used to improve the selection process of leader node [13],
 - (c) The HotStuff is used to reach the framework of finality, which is integrated into the PoA so as to introduce the block finality mechanism to ensure the absolute security of the block [14].

The contributions of this paper are as follows.

- (1) Some typical application schemes of blockchain technology, edge computing, and traditional data disaster backup are traced back, and their technical characteristics and shortcomings are analyzed.
- (2) Combining HE technology with AES, a new HE-AES encryption algorithm is proposed. Compared with traditional encryption algorithms, it can not only greatly improve the encryption and decryption speed but also resist violent cracking of the adversary.
- (3) The Kademlia algorithm is embedded in the edge server. In the edge computing environment, a microgrid data disaster backup scheme based on blockchain is designed, which can realize the distributed storage and the disaster backup of data automatically.
- (4) The traditional PoA is improved to a certain extent, which can solve the problem that the node information is easy to be disclosed after multiple rounds of consensus and cannot adapt to BFT.
- (5) The security analysis can show that the scheme is safe and reliable.
- (6) The performance evaluation shows the reliability, efficiency, and practicability of this scheme.

2. Related Work

At present, most of the disaster backup schemes of data are based on a centralized technology or cloud backup. The

security of these schemes depends on the third party platform. Once the third party platform is attacked, the data will be stolen or lost. There are a few of data disaster backup schemes based on blockchain technology and edge computing, so the related research mainly focuses on blockchain technology, edge computing, and their applications in data disaster backup schemes.

As far as the application of blockchain technology is concerned, Liu et al. [15] proposed a security scheme based on blockchain to protect vehicle interaction in electric vehicle cloud computing and edge computing based on distributed consensus. Electric vehicles were equipped with a variety of intelligent applications in the network. At the same time, there were a lot of information and energy interactions between electric vehicles and between electric vehicles and infrastructure. The introduction of blockchain can enhance security and privacy. However, the delay was long and the connection was unstable due to the long distance between the cloud center and the vehicle. Kang et al. [16] designed a data sharing scheme based on reputation in vehicle edge network by using blockchain and smart contract. The triple subjective logic model was used to manage the reputation of vehicles accurately, which can realize accurate reputation management of high-quality data sharing among vehicles. However, the mining task was needed in the process of block generation, that was Proof of Work (PoW), which needed a lot of computing resources and energy, so it was difficult to widely promote blockchain in wireless networks. Ma et al. [17] proposed a trusted data management scheme based on blockchain in edge computing to protect sensitive data. Before the transaction payload was stored in the blockchain system, the user-defined encryption of sensitive data was designed, and the conditional access and decryption query of protected transactions were realized by using smart contract. However, it is difficult to estimate the computing overhead due to the complexity of encryption and decryption of sensitive data.

As far as the application of edge computing is concerned, Guan et al. [18] proposed an energy Internet data exchange architecture considering the efficiency of edge computing and data security. The edge computing was applied to solve the challenges related to data exchange and data security at the same time. However, the model cannot reflect the actual structural characteristics of the system due to the lack of topological structure caused by complete formulation. Han and Xiao [19] proposed a nontechnical loss (NTL) detection scheme based on edge computing and big data analysis tools to solve the problem of big data NTL fraud detection in smart grid, which provided experience for the development of big data security schemes in smart grid. However, it only focused on the topological connection relationship, and the data interaction relationship was too conceptual to correspond with the actual system components. Rahman et al. [20] proposed an AI-based edge service composition model for edge networks, which was used to find AI rules of functional relationships between edge services dynamically and construct a set of possible composite service plans. In addition, based on the service selection of Skyline optimization method, an AI service composition framework based on privacy protection was proposed, which used Skyline

optimization technology and encrypted data to find the best composite service. However, the default edge device was static in the edge service composition model, but in the practical application process, the mobility of edge devices in the synthesis period needed to be considered, so the practical application of the model was still difficult.

With regard to the application of blockchain technology and edge computing in data disaster backup scheme, Bae and Shin [21] proposed an automatic recovery system using blockchain, assuming that the copy file was created and managed as a block successfully. If the copy was destroyed, the disaster recovery system would check the authenticity of the copy file by pointing to the blockchain and then continue to recover. However, the PoW was adopted in the system, so the transaction processing efficiency was low, which cannot meet the real application scenarios. Su et al. [22] proposed a secure content caching scheme for disaster backup of mobile social networks based on fog computing, which introduced scrambling and partitioning methods to encrypt the content, delivered and stored the encrypted content in multiple cloud servers, and developed an auction game model to select the optimal cloud server. The edge node and the selected server can achieve the largest utility. However, the plaintext was only scrambled without any encryption or hiding in this scheme, so its security needed to be improved. Lao et al. [23] proposed a disaster backup scheme of power data based on blockchain technology, which can guarantee the consistency, non-tampering, and confidentiality of backup data effectively by using all or nothing transform technology and threshold secret sharing algorithm and greatly avoid the risk of single point failure and enhance scalability. However, the original data cannot be recovered when the number of fault nodes exceeded a certain number in the scheme. Liang et al. [24] proposed a secure data backup and recovery scheme in the industrial blockchain network environment, which realized the recovery code with high precision and maintainability in industrial network 4.0. In addition, a unique and relevant backup structure including data consensus and smart contract can be used for fast local code backup of adjacent data backed up in blockchain-based networks. However, the performance of data backup and recovery of multiple nodes was not high, and the real-time and security of data backup and recovery were improved at the expense of data backup rate.

In summary, although the current research on data disaster backup schemes is very extensive, there are some shortcomings more or less. These problems mainly focus on centralized backup, low transaction efficiency, poor security, and so on. Therefore, in view of the inability of automatic disaster backup of data, low data processing efficiency, and poor security in microgrid, a blockchain-based microgrid data disaster backup scheme is designed in the edge computing environment. While automatic disaster backup of data is realized, the efficiency of data processing is ensured, and the security of the data processing is improved at the same time.

3. Scheme Framework

The scheme is mainly composed of microgrid node, registration center, edge server, and blockchain. Therefore, it can

be divided into four layers, which are data layer, supervision layer, storage layer, and consensus layer. The architecture of the scheme is shown in Figure 1. The main symbols used are shown in Table 1. The functions of each part are as follows.

- (1) The microgrid nodes are mainly included in data layer, which are the source of data, and mainly complete the collection and upload of data.
- (2) The registration center is mainly included in supervision layer, which is responsible for generating and distributing passwords and keys for microgrid nodes and edge servers during the initialization. In addition, a node ID is assigned for each edge server randomly by registration center.
- (3) The edge servers are mainly included in storage layer, which are the key of the scheme. The edge server is used to clean, compress, and encrypt the data. Then, the encrypted data block is stored in the edge server. Finally, the hash summary of the encrypted data block is calculated. The hash summary and other information are uploaded to the blockchain for storage.
- (4) The blockchain is mainly included in consensus layer, which is mainly responsible for receiving and storing the effective information uploaded by the edge server. When data recovery is needed, the hash summary of the encrypted data block is transmitted to the edge server.

4. Scheme Design

The blockchain can be divided into public chain, consortium chain, and private chain according to its degree of decentralization [25]. When the blockchain is applied to the microgrid, each node needs to be authorized to join or exit. Otherwise, the excessively open system allows each node to join or exit freely, which will bring chaos to the system and make the management more difficult. Thus, it is more appropriate to choose consortium chain to build the system.

With the development of quantum computing, both symmetric and asymmetric encryption algorithms cannot resist violent cracking, and they are no longer secure absolutely. Consequently, the HE technology is introduced in this scheme, and combining it with the AES, a new algorithm HE-AES is proposed that can resist the violent attack. The plaintext is mapped to seed space through distribution transforming encoder (DTE) to get seed by using HE technology, exclusive-or the seed and password, and symmetric encryption algorithm is used to get ciphertext [26]. When the attacker uses the wrong key to decrypt, the traditional encryption algorithm will return a string of messy codes. It is easy for the attacker to determine the key is wrong. If the HE technology is used, the attacker will obtain seemingly reasonable plaintext in order to deceive the attacker [27]. Therefore, the attacker will fall into the dead cycle of trying the key to obtain the fake message constantly, so that the plaintext cannot be recovered.

FIGURE 1: Scheme framework.

TABLE 1: Description of main symbols.

Number	Symbols	Description
1	ID	The identity of the edge server
2	k	Password
3	K	Seed key
4	m	The number of other nodes contained in each K-bucket
5	n	
6	M	The number of edge servers
7	s	Plaintext
8	R	Seed
9	C	Random string
10	N	Ciphertext
11	K_{pu}	The number of consensus nodes
12	K_{pr}	Public key
13	x	Private key
14	y	Random input
15	π	Random output
16	type	Zero-knowledge proof
17	round	The type of node
18	τ	The consensus stage
19	w	The number of node winning sortition
20	W	Weight
21	j	Total weight
22	d	The result of sortition
23	a	Damping coefficient
24	b	The number of incoming links for the node
25	c	The number of outgoing links for the node
26	I	The total number of links to the node
27	Q	The incoming link reputation of the node
28	P_{BG}	The outgoing link reputation of the node
29	l	The block generation probability of the node
30	p_m	The size of the seed space
31	η	The distribution of the message space
32	λ	Message space
33	p_k	The maximum message probability
		The distribution of the password

4.1. Initialization. Four tasks are mainly completed by the system during the initialization, which are the registration of microgrid nodes, distribution of passwords and keys, access of

microgrid nodes to the edge server, and distribution of ID for the edge server. The initialization sequence diagram is shown in Figure 2. The specific flow of the process is as follows:

FIGURE 2: Initialization sequence diagram.

- (1) The user's private information (ID number, mobile phone number, house number, etc.) is sent to the registration center by microgrid node.
- (2) The validity of the node information is verified by registration center. If it is legal, a pair of passwords k and a pair of seed keys K will be generated. Otherwise, an error will be returned.
- (3) A password k and a seed key K are returned to the microgrid node by the registration center, and the user can create an account with them.
- (4) The user sends the account information to the adjacent edge server and downloads the latest information from the edge server to connect the microgrid node to the edge server.
- (5) The received information of microgrid node account is uploaded to the registration center by the edge server.
- (6) The information of microgrid node account is searched and verified by the registration center. If it is accurate, the ID will be assigned to the edge server. Otherwise, an error will be returned. Since the Kademlia algorithm is embedded in the edge server, the length of ID is 160 bits [28].
- (7) The password k and seed key K that match the microgrid node and edge server ID are sent to the edge server by the registration center.

4.2. Data Backup. In order to solve the problems of poor security and vulnerability to data leakage caused by traditional centralized data backup, the Kademlia algorithm is embedded in the edge server, which can realize the distributed backup of data. The hash value of 160 bits on encrypted data is taken as the key, and the encrypted data is taken as the value in the algorithm. Then, the encrypted data is backed up in the form of key-value pairs on multiple edge servers close to the ID. The Kademlia network can accommodate up to 2^{160} nodes [29], and its backup capacity is much larger than the number of devices in the actual network. Therefore, it can meet the scalability requirements of large-scale microgrid applications [30].

In the Kademlia network, only a part of encrypted data is backed up in each edge server. For the calculation of node distance, the Kademlia algorithm is realized by exclusive-or operation of two node IDs, and then, the operation result is converted into decimal, which is the distance between two nodes, such as

$$01010000 \oplus 01010110 = 00000110. \quad (1)$$

Converting the result of (1) to decimal is 6, which means that the distance between two nodes is 6. There are 160 layers of K-bucket list in each edge server, which can back up the node information within the range $[2^i, 2^{(i+1)})$ ($i \in [0, 160)$) of its distance. m other nodes information is backed up in each K-bucket. This backup method can make n edge servers need $\lg n$ queries at most to find the required information.

The distributed backup structure in edge computing is shown in Figure 3.

In the Kademlia algorithm, there are four instructions in each node as follows:

- (1) PING: verification. Check whether the remote node is online.
- (2) STORE: storage. Store resources in the form of key-value pairs on a node.
- (3) FIND_NODE: search node. Search m nodes near the target node.
- (4) FIND_VALUE: search resources. Search the value of ID node close to the key according to the given key.

When a new node joins the Kademlia network, a seed node is needed as an intermediary. The process of new node joining the network is shown in Figure 4, and the specific steps are as follows.

Step 1. The seed node is joined into the K-bucket of the new node to make it the seed node of the new node.

Step 2. A query request FIND_NODE is sent to the seed node, which can indicate that more nodes are searched through the seed node cluster by the new node. Moreover, more other nodes are searched with similar ID to the new node quickly.

Step 3. After receiving the query request from the new node, the new node is added to the K-bucket of the seed node. Then, the information of at most m other nodes closest is returned to the new node.

Step 4. After receiving the response from the seed node, the received node information is added to the K-bucket of the new node.

Step 5. Steps 2, 3, and 4 are repeated by the new node for other new nodes until a detailed routing table is established.

After a series of tasks were completed during the initialization, the microgrid node joined the system and entered the data backup successfully. The pseudocode of data backup algorithm is shown in Algorithm 1, and the specific process of data backup stage is as follows.

- (1) The transaction data is uploaded to the edge server by microgrid node.
- (2) The edge computing is used to preprocess data, remove redundant data, reduce data size, and improve data quality by edge servers.
- (3) The HE-AES algorithm is used to encrypt the preprocessed data. The encryption process is shown in Figure 5, and the pseudocode of encryption algorithm is shown in Algorithm 2. The specific encryption process is as follows:
 - (a) The plaintext M is mapped to the seed space by DTE to obtain the seed s
 - (b) The hash function H and random string R are used to hash and salt the original password k to obtain the password k'

FIGURE 3: Distributed backup structure in edge computing.

- (c) The ciphertext C^* is obtained by exclusive-or operation of the processed password k' and seed s
- (d) The AES is used to encrypt the ciphertext C^* again to get the ciphertext C
- (4) The hash value of encrypted data is calculated and the encrypted data is stored in m edge servers, whose node ID is close to its hash value.
- (5) The hash value of the encrypted data, timestamp, and the information of microgrid node are uploaded to the blockchain for storage.

4.3. Improvement of PoA. For the storage process of data on the chain, in view of the low efficiency of data processing, easy bifurcation, and fast performance degradation of the traditional consensus mechanism used in the blockchain consensus stage, a new and efficient consensus mechanism PoA is adopted to make the nodes reach consensus in the blockchain network. The PoA mainly depends on a group of authorized nodes. For any transaction of unauthorized nodes, the authorized node will implement consensus, that is, the authorized node proposes a new block, and other verification nodes vote to verify it. When the proposed block is confirmed by at least $N/2 + 1$ verification node, the block is finally confirmed, and the whole system reaches consensus [31]. The rotation mode is adopted by the traditional PoA, and the authorized and leader nodes are selected from the consensus nodes through the redundancy method [32]. Although this method is simple and efficient, the information of the authorized node is easy to be disclosed after multiple rounds of consensus, which will cause certain security problems and even lead to a sharp decline in the efficiency of consensus. Therefore, on the basis of the traditional PoA, the following three improvements are made to the PoA.

- (1) The committee-endorsing mechanism is introduced to improve the selection process of authorized nodes. The VRF is used to select authorized nodes from consensus nodes to form a committee.
- (2) PageRank algorithm and Pareto distribution are introduced to improve the selection process of leader

FIGURE 4: New nodes join the network.

```

(1) Data_ID = hash[Enc(data)]
(2) Block_data ← (Microgrid_account, Microgrid node_address, Data_timestamp, Data_ID)
(3) for each  $i \in [0, m]$  do
(4)   Storage Node ← FIND_NODE(Data_ID)
(5)   if PING Storage Node == True then
(6)     STORE (Block_data, Enc(data))
(7)   end if
(8)   Data_ID + = 1
(9) end for

```

ALGORITHM 1: Data backup.

FIGURE 5: HE-AES encryption process.

```

(1)  $s \leftarrow \text{Encode}(M)$ 
(2)  $R \leftarrow \{0, 1\}^n$ 
(3)  $k' \leftarrow H(R, k)$ 
(4)  $C^* \leftarrow k' \oplus s$ 
(5)  $C \leftarrow \text{Enc}_K(C^*)$ 

```

ALGORITHM 2: HE-AES encryption.

node. PageRank algorithm and Pareto distribution are used to select the leader node from the committee.

- (3) The block finality mechanism is introduced to ensure the absolute security of the block. In order to make blocks confirmed by BFT consensus, the status information of blocks in the process of reaching finality is added in each block based on HotStuff consensus framework.

For the selection of authorized nodes, the committee-endorsing mechanism is introduced. The VRF is used to select authorized nodes from consensus nodes to form a committee. The committee-endorsing mechanism is a way to change the existing generation of new blocks essentially. In the process of block generation, committee members need to verify the new block proposal issued by the leader node. After verification, committee members will sign the proposal as their official endorsement. The leader node is required to collect enough endorsements from committee members and then add relevant information to the new block by the consensus algorithm. The VRF is used to ensure that committee members are selected from consensus nodes randomly. The VRF comes with a key generation algorithm, which can generate public key K_{pu} and private key K_{pr} . The private key is used to hash an input x to get a random output y ; that is,

$$y = \text{hash}_{VRF}(K_{pr}, x). \quad (2)$$

In addition, its private key can also be used to construct a zero-knowledge proof π to prove that the y is correct output; that is,

$$\pi = \text{Prove}_{VRF}(K_{pr}, x). \quad (3)$$

Anyone can use π to obtain y by the following calculation:

$$y = \text{hash}_{\text{Prove}_{VRF}}(\pi). \quad (4)$$

The verifier holding the public key K_{pu} can verify whether the output y is equal to the output based on the input x of the private key.

A sortition selection algorithm based on VRF is designed in this scheme. Each selection of the authorized node will not be disclosed, which can ensure the unpredictability and randomness of the authorized node. At the same time, the identity of the authorized node can be verified by the verifying sortition algorithm. By inputting the private key and related parameters, the consensus

node can obtain the random hash value and the winning result of this round of VRF after calculation. The pseudocode of the sortition selection algorithm is shown in Algorithm 3.

After a node has won the sortition, the sortition message will be broadcast, and other nodes will call the verifying sortition algorithm to verify it after receiving the message. The zero-knowledge proof π is first verified, and then, the number of selected child nodes is obtained similar to the sortition process. If the result is 0, it means that it has not been selected, so it is compared with the j broadcast by the node to verify its correctness. The pseudocode of verifying sortition algorithm is shown in Algorithm 4.

For the selection of leader node, the PageRank algorithm is introduced to calculate the reputation for each node in the blockchain network, and the leader node is determined according to the reputation. The PageRank algorithm originated from the Google browser calculating the ranking of web pages based on the number of outbound and inbound links [33], and its mathematical model is as follows:

$$PR(A) = \frac{1-d}{N} + d \sum_{i=1}^n \frac{PR(B_i)}{C(B_i)}, \quad (5)$$

where $PR(A)$ is the link probability of web page A , $PR(B_i)$ is the link probability of web page B_i linked to web page A , $C(B_i)$ is the number of outbound links of web page B_i , N is the total number of all web pages in the network, d is the damping coefficient, $d \in [0, 1]$, and the general value is 0.85.

The P2P network of blockchain is similar to the network. The node is equivalent to a web page, the reputation of the node is equivalent to the link probability of the web page, and the outgoing and incoming link of each node are equivalent to the outbound and inbound link of the web page. Each node in the P2P network of the blockchain can establish up to eight outgoing links and 117 incoming links [34]. Therefore, it is appropriate to apply the PageRank algorithm to the P2P network of the blockchain to calculate the reputation for each node. However, it is unrealistic to only rely on the introduction of the PageRank algorithm to select leader node, because, in the P2P local small network of the blockchain, there may be two or more nodes with the same number of outgoing links and incoming links. Therefore, the calculated reputation of the node will be the same.

In order to solve the problem, on the basis of obtaining the reputation, a block generation probability for each node is allocated by introducing the Pareto distribution in the


```

(1)  $\langle \text{hash}, \pi \rangle \leftarrow \text{VRF}_{K_{pr}}(x \| \text{type} \| \text{round})$ 
(2)  $p \leftarrow \tau / W$ 
(3)  $j \leftarrow 0$ 
(4) while  $(\pi / 2^{\text{len}(y)}) \notin (\sum_{q=0}^j B(q; w, p)), \sum_{q=0}^{j+1} B(q; w, p)$  do
(5)    $j++$ 
(6) end while
(7) return  $(\text{hash}, \pi, j)$ 

```

ALGORITHM 3: Sortition selection.

```

(1) if  $\text{VerifyVRF}_{K_{pu}}(\text{hash}, \pi, x \| \text{type} \| \text{round}) == 0$  then
(2)   return 0
(3) else
(4)    $p \leftarrow \tau / W$ 
(5)    $j \leftarrow 0$ 
(6) while  $(\pi / 2^{\text{len}(y)}) \notin (\sum_{q=0}^j B(q; w, p)), \sum_{q=0}^{j+1} B(q; w, p)$  do
(7)    $j++$ 
(8) end while
(9) return  $j$ 

```

ALGORITHM 4: Verifying sortition.

power-law distribution in this scheme. The multiplication of the reputation and block generation probability is used as the actual reputation of each node, which can avoid the appearance of nodes with the same reputation in the local small network successfully. Both normal distribution and power-law distribution are two common data distribution patterns [35], in which one of the typical distributions of the former is the height distribution of human beings, while one of the typical distributions of the latter is the distribution of social wealth. In the P2P network of the blockchain, a small number of nodes occupy the vast majority of the computing power of the entire network, which is similar to the fact that a small number of people own the vast majority of the wealth of the society. Therefore, it is appropriate to introduce Pareto distribution to solve this problem.

Before introducing Pareto distribution, the reputation of incoming links and outgoing links of each node is quantified by (6) and (7), where a is the number of incoming links of node A , b is the number of outgoing links of node A , and c is the sum of the number of links established by node A , $c = a + b$.

$$PR(A_1) = \frac{a}{c}(A), \quad (6)$$

$$PR(A_0) = \frac{b}{c}PR(A). \quad (7)$$

Suppose that the incoming link reputation of a node is $\{I_1, I_2, \dots, I_n\}$, outgoing link reputation is $\{O_1, O_2, \dots, O_n\}$, the former obeys the Pareto distribution of parameters γ and α , and the latter obeys the Pareto distribution of parameters γ and β . The probability distribution functions of the incoming link reputation I and the outgoing link reputation O are expressed as

$$\bar{F}(i) = P(I > i) = \left(\frac{i}{\alpha}\right)^{-\gamma}, \quad i \geq \alpha \geq \gamma, \quad (8)$$

$$\bar{F}(o) = P(O > o) = \left(\frac{o}{\beta}\right)^{-\gamma}, \quad o \geq \alpha \geq \gamma.$$

Then, the probability of node block generation P_{BG} is expressed as

$$P_{BG} = \bar{F}(i)\bar{F}(o) = \left(\frac{io}{\alpha\beta}\right)^{-\gamma}. \quad (9)$$

Therefore, the mathematical model for calculating the reputation of each node in the blockchain P2P network can be expressed as

$$PR(A) = \left[\frac{1-d}{N} + d \sum_{i=1}^n \frac{PR(B_i)}{C(B_i)} \right] \left(\frac{io}{\alpha\beta} \right)^{-\gamma}. \quad (10)$$

When the transaction data is stored on the chain, the reputation for each node is calculated in the current network by using (10), and the blocks are packed and organized by the node with high reputation.

The absolute security guarantee is given to legal blocks by the block finality mechanism. Once a block has been finally achieved, the consensus algorithm can ensure that it cannot be modified, replaced, or removed in the ledger. Even when the system encounters extremely asynchronous network conditions, the consensus algorithm can also guarantee its security.

In order to achieve finality, a block must be confirmed by BFT consensus. The HotStuff consensus mechanism is used to reach a final framework. The framework divides the BFT consensus process into three consecutive stages, and each stage requires more than 2/3 of the nodes to reach

consensus. In order to integrate the framework into PoA, the state information of the block in the process of achieving finality is added to each block. When a committee member endorses a block, the member also confirms the relevant information contained in the block. Therefore, when the subsequent blockchain of a block contains more than 2/3 consensus nodes (as leader node or committee members), the block completes a stage of BFT consensus. In the process of achieving BFT, more than 2/3 nodes to respond online at the same time will not be required, but only the leader node and committee members can respond in time. Such a mechanism can reduce the service delay or temporary unavailability caused by the node's failure to respond in time. The improved PoA consensus process is shown in Figure 6, including the following four stages:

- (1) *Leader Selection*. First, (2) and (3) are used to perform five rounds of calculation. At the same time, the sortition selection algorithm is used to select five authorized nodes to form the committee, and the verifying sortition algorithm is used to verify the result of sortition by other consensus nodes. Then, (11) is used to calculate the reputation for the authorized nodes in the current committee. The node with high reputation is selected as the leader node. Other nodes are named as validate node 1, validate node 2, validate node 3, and validate node 4, respectively.
- (2) *Block Proposal*. The leader node packages the new transaction request, constructs a new block, signs the block data and hash, and then broadcasts the block and signature to all authorized nodes in the network.
- (3) *Block Acceptance*. After receiving the message from the leader node, the validate node verifies the validity of the block, signs the block, and returns the signature to the leader node.
- (4) *Block Confirmed*. The leader node can submit the block only after receiving at least 2/3 of the valid signatures of the nodes and broadcast the new block in the network finally.

4.4. Data Recovery. When the microgrid node is damaged by human factors or irresistible natural factors resulting in data loss, a data recovery request can be sent to the nearest edge server by the microgrid node. The pseudocode of data recovery algorithm is shown in Algorithm 5, and the recovery process is shown in Figure 7. The specific data recovery steps are as follows.

Step 1. The data recovery request including account information and timestamp is sent to the nearest edge server by the microgrid node.

Step 2. The data recovery request information submitted by the microgrid node is used to query the microgrid node information on the blockchain by the edge server.

Step 3. The microgrid node information is verified by the blockchain. If the verification fails, an error is returned. Otherwise, Step 4 will be executed.

Step 4. The hash value of the required data is searched and returned to the edge server by the blockchain.

Step 5. The hash value of the data is used as the key by the edge server, and the Kademlia algorithm is used to search for the matching encrypted data automatically.

Step 6. The hash value of the encrypted data is calculated and compared with the hash value returned from the blockchain by the edge server. If the two values are different, it will return the warning of data damage to the microgrid node. Otherwise, Step 7 will be executed.

Step 7. The encrypted data is returned to the microgrid node.

Step 8. The HE-AES algorithm is used to decrypt the data to obtain the required data by the microgrid node. The pseudocode of the HE-AES decryption algorithm is shown in Algorithm 6, and the decryption process is shown in Figure 8. The specific decryption process is as follows:

- (a) The AES is used to decrypt the ciphertext C to obtain C^*
- (b) The hash function H and random string R are used to hash and salt the original password k to obtain the password k'
- (c) The seed s is obtained by exclusive-or operation of the processed password k' and ciphertext C^*
- (d) The reverse sampling of DTE is used to decode the seed s to obtain plaintext M

5. Security Analysis

Combining HE technology with AES, a new HE-AES algorithm is proposed, which is used to encrypt data, so as to ensure the security of data.

Suppose that, for an attacker A who tries to recover a message, if the output message is consistent with the original message through violent cracking, the attacker wins. The advantage $\text{Adv}_{\text{HE-AES}}^{\text{MR}}$ of the attacker A in cracking the HE-AES algorithm can be expressed as

$$\text{Adv}_{\text{HE-AES}}^{\text{MR}}(A) = \Pr(\text{MR}_{\text{HE-AES}}^A \Rightarrow \text{True}), \quad (11)$$

where $\Pr(\text{MR}_{\text{HE-AES}}^A \Rightarrow \text{True})$, is the probability that attacker A recovers the message successfully.

Suppose the seed space size of DTE is l bit, p_m is the distribution of message space η , the maximum message probability is λ , p_k is the distribution of passwords, and the maximum weight is ω , and then, the following can be proved:

$$\text{Adv}_{\text{HE-AES}}^{\text{MR}}(A) \leq \omega(1 + \varepsilon) + \delta, \quad \delta = \frac{1 + \mu}{2^l}, \mu = \frac{1}{\omega}, \quad (12)$$

where

$$\varepsilon = \frac{\mu_0^2}{2\eta_0} + \frac{e\mu_0^4}{27(\eta_0^2 - e\mu_0^2)}, \quad \mu_0 = \frac{3}{\omega}, \eta_0 = \frac{2}{\lambda}. \quad (13)$$

Normally, $\eta_0 \gg \mu_0^2$, so ε will be very small. When the size of the seed space l is very large, δ will approach 0. Therefore,

FIGURE 6: Improved PoA consensus process.

```

(1) key_Data_key
(2) for each  $i \in [0, m]$  do
(3)   Search Node  $\leftarrow$  FIND_VALUE(key)
(4)   if PING Search Node == True then
(5)     GET (Data_ID)
(6)   end if
(7) end for
(8) Block_data  $\leftarrow$  Map (Data_ID)
(9) find (Block_data, Enc (data))
(10) if Data_ID == Block_data.Data_ID then
(11)   return Enc (data)
(12) else
(13)   return Warning
(14) end if

```

ALGORITHM 5: Data recovery.

FIGURE 7: Data recovery process.

- (1) $C^* \leftarrow \text{Dec}_K(C)$
- (2) $R \leftarrow \{0, 1\}^n$
- (3) $k' \leftarrow H(R, k)$
- (4) $s \leftarrow k' \oplus C^*$
- (5) $M_Decode(s)$

ALGORITHM 6: HE-AES decryption.

FIGURE 8: HE-AES decryption process.

$\text{Adv}_{HE-AES}^{MR}(A)$ is close to 0, so the probability of attacker A cracking the algorithm successfully is close to 0, which means that the algorithm is safe.

6. Performance Evaluation

In order to further verify the performance of the scheme, a series of experiments are designed to further illustrate the efficiency of the scheme by comparing with other schemes in terms of backup delay and recovery delay, backup time and recovery time, recovery rate, integrity rate, block propagation delay, computing overhead, block generation time, and transaction throughput. Since these indexes can also show excellent results in some schemes in the small data sample environment, in order to fully explain the data processing efficiency of this scheme, big data is selected as the sample. The data samples are increased from 500 MB to 2500 MB by 500 steps. The automatic recovery system based on blockchain (ARS) [21], disaster backup scheme of mobile social network content supporting fog computing (DBMSN) [22], power data disaster backup scheme based on blockchain (PDDB) [23], and industrial blockchain environment data disaster backup scheme (IBEDB) [24] are mainly selected as comparison scheme. In order to reduce the influence of unstable factors on the experiment, 20 measurements were carried out for each selected data sample, and then, the average value was taken as the index data of the sample.

6.1. Experiment Setup. In this scheme, one host is used as the microgrid node, which is named microgrid node, and the other three hosts are used as edge server nodes, which are named edge node 1, edge node 2, and edge node 3,

respectively. The specific setup information of each device is shown in Table 2.

6.2. Logic Realization Based on Smart Contract. The smart contract on the blockchain is an executable program in a sandbox environment [36]. After triggering preset conditions, it can execute and complete the set goals automatically. The smart contract is used to realize the information interaction between the edge server and the blockchain in the scheme. The initialization parameters, data pre-processing, data encryption, ciphertext storage, data block storage, query of microgrid node information, authority verification, hash value verification, and data recovery are included in the process of disaster backup scheme. The pseudocode of smart contract is shown in Algorithm 7.

6.3. Evaluation Results. The safe and efficient disaster backup of microgrid data is realized in the edge computing environment. Therefore, the typical disaster backup schemes in recent years (including ARS, DBMSN, PDDB, and IBEDB) are selected as the reference in the experiment and compared with this scheme to further illustrate the data processing efficiency.

6.3.1. Backup and Recovery Delays. The change of backup delay and recovery delay is observed by adjusting the data size in the experiment. The experimental results of backup delay and recovery delay are shown in Figure 9.

The backup delay and recovery delay of all schemes are increased with the increase of data size. When the data size is up to 2500 MB, the ARS has the worst effect in dealing with big data, and this scheme has the best effect. The backup

TABLE 2: Experimental equipment configuration information.

Device	CPU	RAM (GB)	OS
Microgrid node	Intel Core i7-6870HQ (2.7 GHz)	32	Ubuntu 20.04.1 LTS
Edge node 1	Intel Xeon E3-1220 (3.0 GHz)	32	Ubuntu 20.04.1 LTS
Edge node 2	Intel Xeon E5-2609 (2.4 GHz)	24	Ubuntu 20.04.1 LTS
Edge node 3	Intel Xeon E5-2620 (2.1 GHz)	24	Ubuntu 20.04.1 LTS

```

(1) Initialized parameters
(2) Data preprocessing using edge computing
(3)  $C \leftarrow \text{Enc}_{HE-AES}(\text{data})$ 
(4) Store  $C$  in edge server
(5) Select authorized nodes using Algorithm 3
(6) Verify authorized nodes using Algorithm 4
(7) Calculate the reputation using (10) for each node
(8) Select leader node to generate block according to the reputation
(9)  $\text{Block\_hash} = \text{hash}(C)$ 
(10)  $\text{Microgrid\_node\_address} = \text{Microgrid\_node\_account.address}$ 
(11)  $\text{Block\_data} \leftarrow (\text{Microgrid\_account}, \text{Microgrid\_node\_address}, \text{data\_timestamp}, \text{Block\_hash})$ 
(12) Send the  $\text{Block\_data}$  to blockchain for storage
(13) Send the  $\text{data\_request}$  to the closest edge server
(14) Integrate the  $\text{data\_information}$  of the  $\text{data\_request}$  into the  $\text{data\_name}$ 
(15)  $\text{Microgrid\_node(request)\_address} = \text{GetMicrogrid\_node\_address}(\text{data\_name})$ 
(16) if  $\text{Microgrid\_node(request)\_address} = \text{Microgrid\_node\_address}$  then
(17)    $\text{hash\_input} = \text{hash}(C)$ 
(18) end if
(19) if  $\text{Block\_data.Block\_hash} == \text{hash\_input}$  then
(20)   Send  $C$  to the  $\text{Microgrid\_node(request)}$ 
(21) else
(22)   return False
(23) end if
(24)  $\text{data} \leftarrow \text{Dec}_{HE-AES}(C)$ 

```

ALGORITHM 7: Smart contract.

FIGURE 9: Experimental results of backup delay and recovery delay. (a) Backup delay. (b) Recovery delay.

delay and recovery delay of the former exceed 1000 ms, while the backup delay and recovery delay of the latter do not exceed 250 ms. The best performance of other three schemes is the DBMSN, and the worst is the PDDb, which is because the fog computing is introduced in DBMSN and its function is similar to edge computing. However, edge computing devices are closer to the data source than fog computing, so edge computing performs better when processing big data. Therefore, the PDDb is not as good as this scheme. The IBEDb is integrated into the cloud storage technology, and the blockchain technology is simply used in PDDb. Cloud storage technology also has certain advantages in processing big data. Therefore, the IBEDb is better than PDDb.

6.3.2. Backup and Recovery Times. The data size is set to 500 MB in this experiment. Under this premise, the backup time (including data encryption and storage) and recovery time (including data download and decryption) of each scheme are tested. The experimental results are shown in Figure 10.

The backup time and recovery time of the ARS is the longest, and the time of this scheme is the shortest. Compared with other comparison schemes, the backup and recovery times of this scheme are reduced by at least 11.1% and 23.2%, respectively. The backup and recovery times of the PDDb are almost the same, because the symmetric encryption algorithm is adopted, and the size of ciphertext and plaintext is the same [37]. The recovery time of the DBMSN and IBEDb is longer than the backup time, because the decryption time is longer than the encryption time. In Figure 10(a), the proposed scheme in DBMSN is the best except this scheme, but in Figure 10(b), the scheme proposed in PDDb is the best except this scheme, which is because the AES is adopted in PDDb, and the encryption time and decryption time are similar, while the partition scrambling method is used to encrypt data in DBMSN, and the decryption time is greater than that of encryption time.

6.3.3. Recovery Rate. Recovery rate refers to the proportion of recovered data in the original data, which can reflect the practicability. In this experiment, the change of recovery rate is observed by adjusting the data size. The experimental results of recovery rate are shown in Figure 11.

The recovery rate of all schemes is decreased with the increase of data size, but the decline rate of this scheme is the slowest. When processing the data below 1000 MB, the advantage of this scheme is not obvious compared with other schemes. However, when dealing with the big data above 1000 MB, the recovery rate of this scheme is obviously higher than that of other schemes. Specifically, when processing 1500 MB, 2000 MB, and 2500 MB data, the recovery rate of this scheme is at least 7.5%, 12.2%, and 15.2% higher than other schemes. When the data size is up to 2500 MB, the data recovery rate of the scheme can reach 52%.

6.3.4. Integrity Rate. Integrity rate refers to the proportion of the recovered complete data in the recovered data when the distributed data storage system is attacked illegally. This

index can reflect the integrity of the recovered data. In this experiment, the change of integrity rate is observed by adjusting the data size. The experimental results of integrity rate are shown in Figure 12.

When the distributed data storage system is attacked illegally, the integrity rate of all schemes is decreased with the increase of data size. The integrity rate of the DBMSN drops the fastest because the traditional centralized backup method is adopted. The performance of ARS and PDDb is similar, the performance of this scheme is the best, and the performance of IBEDb is second only to this scheme, which is because not only does the traditional centralized storage structure change by blockchain technology, but also the automatic backup and recovery of data is realized by the Kademlia algorithm in this scheme. Therefore, when the data size is up to 2500 MB, in the face of illegal attacks, the recovery rate can reach 62.5%.

6.3.5. Block Propagation Delay. The block transmission delay, block verification delay, and node information exchange delay are included in the block propagation delay [38]. The block propagation delay is not only related to the reputation of incoming link and outgoing link but also affected by parameter γ . Therefore, this experiment synthesizes three factors to analyze the influence of incoming link reputation, outgoing link reputation, and parameter γ on block propagation delay. The experimental results of block propagation delay are shown in Figure 13.

In the case of constant parameter γ , the block propagation delay will be decreased with the increase of the reputation on the incoming link and the outgoing link, which is because whether the reputation of the incoming link or the outgoing link increases, the number of nodes will be decreased, and the block verification delay will also be decreased. Therefore, the block propagation delay will be decreased. When the reputation of the incoming link or the outgoing link is constant, the block propagation delay will be decreased with the increase of parameter γ , which is because both the reputation of the incoming link and the outgoing link obey the Pareto distribution of parameter γ , and the reputation will be increased with the increase of parameter γ . According to the previously mentioned analysis, the block propagation delay will be decreased when the reputation increases.

6.3.6. Computing Overhead. In this experiment, the data size is set to 500 MB. Under this premise, the storage time of data is used to measure the computing cost of the system. The experimental results of computing overhead are shown in Figure 14.

The computing overhead of the ARS is the largest, and the storage time is up to 235 ms. The computing overhead of this scheme is the smallest, and the storage time is only 36 ms. Compared with the ARS, DBMSN, PDDb, and IBEDb, the computing overhead of this scheme is reduced by 84.7%, 30.8%, 64.4%, and 56.1%, respectively. This is because the ARS and PDDb both use the traditional Ethereum platform to build the blockchain, while the

FIGURE 10: Experimental results of backup time and recovery time. (a) Backup time. (b) Recovery time.

FIGURE 11: Experimental results of recovery rate.

FIGURE 12: Experimental results of integrity rate.

fragmentation technology is used to realize the distributed storage of data in IBEDb, which can reduce the computing overhead. The fog computing is used to process big data in the DBMSN, and the computing overhead is greatly reduced. The edge computing is used in this scheme, which is more efficient than fog computing, so the computing overhead is minimal.

6.3.7. Block Generation Time. The block size is set to 8 MB in the experiment. Under this condition, the block generation time can be observed by changing the number of blocks, which can reflect the block speed of different schemes directly. Since the DBMSN does not use blockchain technology, it needs to be replaced. Similar to the disaster backup

process, two stages of data storage and download are included in the process of data sharing [39]. Therefore, a data sharing scheme is selected as an alternative scheme in this experiment, that is, the Internet of Things data sharing scheme (DSIoT) [40]. The experimental results of block generation time are shown in Figure 15.

The block generation time of all schemes are increased with the increase of the number of blocks, which is because when the number of nodes is constant, the number of packed blocks is increased, and the block generation time will be increased naturally. The ARS and PDDb have the fastest growth, which is due to the blockchain built on the traditional Ethereum platform. The DSIoT has been improved greatly compared with the previous schemes, because a new consensus mechanism Proof of Concept (PoC) is adopted.

FIGURE 13: Experimental results of block propagation delay. (a) Influence of incoming link reputation and parameter γ on block propagation delay. (b) Influence of outgoing link reputation and parameter γ on block propagation delay.

FIGURE 14: Experimental results of computing overhead.

This scheme has the shortest block generation time and the highest block generation rate, which is because a more efficient PoA than PoC is adopted.

6.3.8. Transaction Throughput. The transaction throughput of blockchain refers to the number of transactions completed per unit time, which is one of the typical evaluation indexes of blockchain [41]. In this experiment, the block size is set to 8 MB, and the throughput is observed by changing the block generation time. The DSIoT is used as the alternative comparison scheme to replace the DBMSN, and the other comparison schemes remain unchanged. The experimental results of throughput are shown in Figure 16.

The throughput of all schemes is decreased with the increase of block generation time, because the increase of block generation time means that the number of transactions processed decreases at the same time. The throughput of the ARS and PDDDB is the lowest, both below 1500 tps,

FIGURE 15: Experimental results of block generation time.

which is because the traditional Ethereum platform is used to build the blockchain without any optimization. The throughput of this scheme is the highest and can be close to 15000 tps, which is because not only the Kademlia algorithm is used to realize the distributed storage of data automatically, but also the consensus mechanism is changed in this scheme, which is the decisive factor affecting the throughput [42]. The efficient PoA is adopted, which can greatly improve the efficiency of data processing. The throughput of the IBEDB is fluctuated in the range of (1000, 10000), which is because the fragmentation processing technology is adopted to optimize the transaction verification, which can improve the throughput slightly. The throughput of the DSIoT is fluctuated in the range of (1500, 15000), and its maximum throughput can be close to 13500 tps, which is because the

FIGURE 16: Experimental results of throughput.

traditional blockchain building platform is changed and the improved permissionless Ethereum platform is adopted to build the blockchain, which can improve the throughput greatly.

7. Conclusion

In view of the problems of traditional data disaster backup schemes, such as overreliance on the third party, inability to implement disaster backup automatically, low security, poor reliability, and low efficiency of data processing, a microgrid data disaster backup scheme based on blockchain is designed in the edge computing environment, which can realize data disaster backup automatically by using Kademlia algorithm. The improved PoA is used to improve the efficiency of data processing greatly, which can solve the problem of traditional centralized disaster backup and improve the security and reliability of the system.

At the same time, the Internet of Things (IoT), Internet of Vehicles (IoV), and Industrial Internet of Things (IIoT) are also facing data security problems. Because its structure is similar to the microgrid, the architecture of this scheme can be integrated into it. Specifically, edge devices can be sunk to the data source, and edge computing is used to process the big data. The Kademlia algorithm is embedded in the edge server to realize the distributed storage and automatic recovery of data, and the improved PoA is adopted to make each node reach consensus and pack the blocks, which can improve the efficiency of data processing. Therefore, a new idea to solve the problem of data disaster backup in the field of IoT, IoV, and IIoT is provided by this scheme.

Although edge computing is used to preprocess the big data of microgrid and reduce the data size in the scheme, the power consumption of the server is increased undoubtedly by encrypting and storing the preprocessed data in the local edge server. Therefore, the future research direction is to

reduce the power consumption of the edge server as much as possible on the premise of ensuring the security and efficiency of the scheme. We can consider storing the encrypted data in the distributed database or using cloud storage technology to store it in the cloud.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to acknowledge East China Jiaotong University for the lab facilities and necessary technical support. This research was funded by the National Natural Science Foundation of China under grant no. 61563016 and Science and Technology Project Foundation of Jiangxi Provincial Education Department under grant no. GJJ14371.

References

- [1] Y. Du, J. Wu, S. Li, C. Long, and I. C. Paschalidis, "Distributed MPC for coordinated energy efficiency utilization in micro-grid systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1781–1790, 2019.
- [2] A. Aydeger, N. Saputro, K. Akkaya, and S. Uluagac, "SDN-enabled recovery for Smart Grid teleprotection applications in post-disaster scenarios," *Journal of Network and Computer Applications*, vol. 138, pp. 39–50, 2019.
- [3] T. Rausch, A. Rashed, and S. Dustdar, "Optimized container scheduling for data-intensive serverless edge computing," *Future Generation Computer Systems*, vol. 114, pp. 259–271, 2021.
- [4] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: a survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.
- [5] H. Huang, P. Zhu, F. Xiao, X. Sun, and Q. Huang, "A blockchain-based scheme for privacy-preserving and secure sharing of medical data," *Computers & Security*, vol. 99, Article ID 102010, 2020.
- [6] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for IoT systems," *Cluster Computing*, vol. 23, no. 3, pp. 2067–2087, 2020.
- [7] A. Arivarasi and P. Ramesh, "An improved source location privacy protection using adaptive trust sector-based authentication with honey encryption algorithm in WSN," *Journal of Ambient Intelligence Humanized Computing*, to be published.
- [8] J. Zhong and X. Xiong, "Data security storage method for power distribution internet of things in cyber-physical energy systems," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6694729, 15 pages, 2021.
- [9] L. Zheng, X. Helu, M. Li, and H. Lu, "Automatic discovery mechanism of blockchain nodes based on the Kademlia algorithm," in *Proceedings of the International Conference on Artificial Intelligence and Security*, pp. 605–616, Springer, New York, NY, USA, July 2019.

- [10] Y. Xu, G. Wang, J. Yang, J. Ren, Y. Zhang, and C. Zhang, "Towards secure network computing services for lightweight clients using blockchain," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2051693, 12 pages, 2018.
- [11] R. Gala, E. Shukla, N. Kamble, R. Vijayaraghavan, and D. Patel, "Blockchain-based approach to foster student engagement on campus," arXiv preprint arXiv: 2105.12504, 2021.
- [12] H. Wang and W. Tan, "Block proposer election method based on verifiable random function in consensus mechanism," in *Proceedings of the IEEE International Conference on Progress in Informatics and Computing (PIC)*, pp. 304–308, IEEE, Shanghai, China, December 2020.
- [13] L. Tortosa, J. F. Vicent, and G. Yeghikyan, "An algorithm for ranking the nodes of multiplex networks with data based on the PageRank concept," *Applied Mathematics and Computation*, vol. 392, Article ID 125676, 2021.
- [14] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus in the lens of blockchain," arXiv preprint arXiv: 1803.05069, 2018.
- [15] H. Liu, Y. Zhang, and T. Yang, "Blockchain-Enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.
- [16] J. Kang, R. Yu, X. Huang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.
- [17] M. Zhaofeng, W. Xiaochang, D. K. Jain, H. Khan, G. Hongmin, and W. Zhen, "A blockchain-based trusted data management scheme in edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2013–2021, 2020.
- [18] Z. Guan, Y. Zhang, G. Si et al., "ECOSECURITY: tackling challenges related to data exchange and security: an edge-computing-enabled secure and efficient data exchange architecture for the energy internet," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 61–65, 2019.
- [19] W. Han and Y. Xiao, "Edge computing enabled non-technical loss fraud detection for big data security analytic in Smart Grid," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1697–1708, 2020.
- [20] M. S. Rahman, I. Khalil, M. Atiquzzaman, and X. Yi, "Towards privacy preserving AI based composition framework in edge networks using fully homomorphic encryption," *Engineering Applications of Artificial Intelligence*, vol. 94, Article ID 103737, 2020.
- [21] S. Bae and Y. Shin, "An automated system recovery using block chain," in *Proceedings of the International Conference on Ubiquitous and Future Networks*, pp. 897–901, IEEE, Prague, Czech Republic, July 2018.
- [22] Z. Su, Q. Xu, J. Luo, H. Pu, Y. Peng, and R. Lu, "A secure content caching scheme for disaster backup in fog computing enabled mobile social networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4579–4589, 2018.
- [23] W. Lao, B. Wang, R. Zhang, and J. Wang, "Disaster tolerance and backup scheme of power data based on blockchain," *Computer Engineering*, vol. 45, no. 11, pp. 9–15, 2019.
- [24] W. Liang, Y. Fan, K.-C. Li, D. Zhang, and J.-L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6543–6552, 2020.
- [25] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.
- [26] E. O. Abiodun and A. Jantan, "Modified honey encryption scheme for encoding natural language message," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1871–1878, 2019.
- [27] E. O. Abiodun, A. Jantan, O. I. Abiodun, and H. Arshad, "Reinforcing the security of instant messaging systems using an enhanced honey encryption scheme: the case of WhatsApp," *Wireless Personal Communications*, vol. 112, no. 4, pp. 2533–2556, 2020.
- [28] Y. Hassanzadeh-Nazarabadi, A. Kp, and . zkasapzkasap, "Interlaced: fully decentralized churn stabilization for skip graph-based DHTs," *Journal of Parallel and Distributed Computing*, vol. 149, pp. 13–28, 2021.
- [29] R. Hasimoto-Beltran, F. D. A. Lopez-Fuentes, and M. Vera-Lopez, "Hierarchical P2P architecture for efficient content distribution," *Peer-to-Peer Networking and Applications*, vol. 12, no. 4, pp. 724–739, 2019.
- [30] D. B. Avancini, J. J. P. C. Rodrigues, S. G. B. Martins, R. A. L. Rablo, J. Al-Muhtadi, and P. Solic, "Energy meters evolution in smart grids: a review," *Journal of Cleaner Production*, vol. 217, pp. 702–715, 2019.
- [31] Q. Chen, G. Srivastava, R. M. Parizi, M. Aloqaily, and I. A. Ridhawi, "An incentive-aware blockchain-based solution for internet of fake media things," *Information Processing & Management*, vol. 57, no. 6, p. 102370, 2020.
- [32] O. Samuel, N. Javaid, M. Awais, Z. Ahmed, M. Imran, and M. Guizani, "A blockchain model for fair data sharing in deregulated smart grids," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, Waikoloa, HI, USA, December 2019.
- [33] X. Yin, X. Hu, Y. Chen, X. Yuan, and B. Li, "Signed-page rank: an efficient influence maximization framework for signed social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, 2021.
- [34] W. Ao, S. Fu, C. Zhang, Y. Huang, and F. Xia, "A secure identity authentication scheme based on blockchain and identity-based cryptography," in *Proceedings of the IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET)*, pp. 90–95, IEEE, Beijing, China, August 2019.
- [35] Y. Zhang, Z. Jin, and S. Mirjalili, "Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models," *Energy Conversion and Management*, vol. 224, Article ID 113301, 2020.
- [36] S. Xuan, L. Zheng, I. Chung et al., "An incentive mechanism for data sharing based on blockchain with smart contracts," *Computers & Electrical Engineering*, vol. 83, Article ID 106587, 2020.
- [37] S. Elkana Ebinazer, N. Savarimuthu, and S. Mary Saira Bhanu, "ESKEA: enhanced symmetric key encryption algorithm based secure data storage in cloud networks with data deduplication," *Wireless Personal Communications*, vol. 117, no. 4, pp. 3309–3325, 2020.
- [38] Y. Shahsavari, K. Zhang, and C. Talhi, "A theoretical model for block propagation analysis in bitcoin network," *IEEE Transactions on Engineering Management*, to be published, 2020.
- [39] J. Wang, P. Zheng, Y. Lv, J. Bao, and J. Zhang, "Fog-IBDIS: industrial big data integration and sharing with fog computing for manufacturing systems," *Engineering*, vol. 5, no. 4, pp. 662–670, 2019.

- [40] A. Manzoor, A. Braeken, S. S. Kanhere, M. Ylianttila, and M. Liyanage, "Proxy re-encryption enabled secure and anonymous IoT data sharing platform based on blockchain," *Journal of Network and Computer Applications*, vol. 176, Article ID 102917, 2021.
- [41] J. Yun, Y. Goh, and J.-M. Chung, "DQN-based optimization framework for secure sharded blockchain systems," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 708–722, 2021.
- [42] B. Cao, Z. Zhang, D. Feng et al., "Performance analysis and comparison of PoW, PoS and DAG based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020.