A Novel CP-ABE Based Sidechain Protocol for Distributed Power System Data Storage Management with the Blockchain

Xianzhou Gao
State Grid Key Laboratory of Information & Network Security, Global Energy Interconnection Research Institute Co., Ltd., Nanjing, China, gaoxianzhou@geiri.sgcc.com.cn, Corresponding author

Xiuli Huang
State Grid Key Laboratory of Information & Network Security, Global Energy Interconnection Research Institute Co., Ltd., Nanjing, China
huangxiuli@geiri.sgcc.com.cn

Wei Zhang
Changzhou Power Supply Branch, State Grid Jiangsu Electric Power Co., Ltd. Changzhou, China
910675169@qq.com

Ruxia Yang
State Grid Key Laboratory of Information & Network Security, Global Energy Interconnection Research Institute Co., Ltd., Nanjing, China
yangruxia@geiri.sgcc.com.cn

ABSTRACT

Due to technical bottlenecks such as systems vulnerable to attacks and data are easily tampered with, and it is difficult for the distributed power system data storage management to achieve multi-agent integration and development. Known as the "trust machine," the Blockchain has an essential role in promoting the energy Internet construction with its distributed storage, tamper-proof, traceable, and multi-node sharing technical characteristics. Its technical advantages have great potential application value in the power industry. Most of the existing blockchain schemes directly combine the CP-ABE algorithm with the Blockchain. However, due to the lack of permission inheritance, these schemes have permission to redundancy and waste computing resources. In this paper, a new user-controlled on-chain permission management scheme is proposed. The original CP-ABE model is extended in this scheme, and the concept of role assignment and attribute tree is proposed. The attribute tree uses the multi-tree structure to realize the attribute set's specification description. The connectivity between roles in the attribute tree represents the inheritance relationship of permission so that the permission inheritance and permission granting can be realized. In this paper, a sidechain prototype system is developed for evaluating the scheme, and its performance is tested and analyzed. The experimental results show that the prototype system can realize the user-controlled permission assignment and permission inheritance function of data owner on the Blockchain.

CCS CONCEPTS

• **Computing methodologies** → Modeling and simulation; Model development and analysis; Modeling methodologies.

CONFCDS 2021, January 28–30, 2021, Stanford, CA, USA

ACM ISBN 978-1-4503-8957-0/21/01…$15.00
https://doi.org/10.1145/3448734.3450799

KEYWORDS

blockchain, sidechain protocol, data storage management, CP-ABE

ACM Reference Format:
Xianzhou Gao, Xiuli Huang, Wei Zhang, and Ruxia Yang. 2021. A Novel CP-ABE Based Sidechain Protocol for Distributed Power System Data Storage Management with the Blockchain. In *The 2nd International Conference on Computing and Data Science (CONFCDS 2021), January 28–30, 2021, Stanford, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3448734.3450799

1 INTRODUCTION

Power grid big data refers to the massive multi-source heterogeneous data generated in each smart grid system link, which has large scale characteristics, multiple types, sparse value, and rapid change [1]. In recent years, to cope with global energy problems, countries worldwide have researched smart grids [2]. The establishment of a smart grid covering the power system's entire production process is the power system's primary task. Grid big data is the foundation that supports the safe and stable operation of smart grids. However, big grid data has become a research topic that academia and the power industry are concerned about nowadays. There has been applied in many power production links and has vast application prospects. The research on grid big data is mainly divided into two aspects: First, because the big grid data comes from the massive smart meters, sensors, electrical equipment, and other equipment in the smart grid, it has a large volume, complex type fast speed. How to efficiently manage it is a big problem in the research process of grid big data; second, big grid data has high utilization value. The grid data analysis can be used in the operation, operation, and supply of the grid. It can be used in various aspects and levels on the side transmission side and demand side. For example, it can predict photovoltaic and wind power generation's output power, improve the grid's ability to accept photovoltaic and wind power, optimize the operation mode and flow of the grid, and learn the electricity consumption rules of power users. , Reasonably set up the power demand response system, and even optimize the internal management structure of power enterprises and so on [3]. However, the prerequisite for applying grid big data should be the efficient collection, storage, and management of grid big data. The

power system still has many problems in the storage and management of grid big data: data definitions are not uniform, and data storage contents are inconsistent. The low quality of data content has brought substantial obstacles to grid big data in power systems.

Grid big data is large in volume, diverse in types, widely distributed in data sources, and extremely fast [4]. Continued use of traditional centralized data management methods is inefficient and low in security. Simultaneously, it will seriously affect the value mining of grid big data by enterprises [5]. The emergence of blockchain technology provides a new technical path for the digital asset management of big data in the power grid. The Blockchain has a distributed consensus mechanism, chain block structure, and asymmetric encryption algorithm. So Blockchain has the advantages of decentralization, trustlessness, information traceability, and information difficult to tamper. These advantages are suitable for grid big data management. The requirements are very consistent, so corresponding solutions can be developed based on blockchain technology to solve the problems [6].

The traditional distributed power system data storage management technology mainly relies on a trusted third party to complete authority management [7]. The data owner cannot directly control the process of granting data access authority, and there is a particular risk of data leakage. With the continuous development of blockchain technology, some researchers have taken advantage of the decentralized characteristics of Blockchain and proposed a user-controlled authority management scheme without a trusted third-party platform. El-Hindi et al. [8] combined the traditional database layer with the storage layer constructed using blockchain technology and used standard access interfaces and data management technology for data sharing. Wang et al. [9] used the Ethereum blockchain and attribute-based encryption technology to build a new distributed file storage system, applied to distributed storage systems with higher fine-grained requirements storage. The single point of failure that often occurs in cloud storage systems is highly resistant. Guy Zyskind et al. [10] proposed a private data storage scheme that mixes on-chain storage and off-chain storage. This scheme can guarantee users' control over their data when a third-party platform wants to access user-stored data. At times, transactions need to be performed on the Blockchain, so users can have the right to know the access behavior and the use of their data. Yingying Yao et al. [11] proposed an identity management service architecture based on an attribute-based encryption algorithm and permission chain. Compared with the traditional identity management service architecture, this architecture is characterized by reduced computing resources and storage resources, which makes it applicable to in-vehicle cloud computing. Ouaddah et al. [12, 13] proposed a blockchain-based access control architecture, which uses blockchain smart contracts to implement token issuance and then access control. However, this architecture requires the data owner's permission when the token expires or when a new access request occurs, so the time cost is high. Pianno et al. [14] proposed an improved access control design based on this architecture. By designing the storage of content on the chain, the time cost is improved to satisfy the IoT environment requirement.

The existing blockchain-based data storage management implementation schemes are usually combined with the CP-ABE algorithm to achieve more fine-grained rights management. The data owner can generate the corresponding decryption key according to the data user's attributes and assign the information's rights. Encrypted by a specific access control structure, data users who meet the attribute requirements can obtain permission distribution information through the Blockchain to obtain data access permissions. However, the existing rights management scheme that combines CP-ABE and blockchain technology has authorization redundancy. There will be a problem of high management complexity in enterprise-level application scenarios due to the excessive number of users. The actual use process will cause a tremendous computing burden to the data owner and waste computing resources. In response to the above problems, this paper proposes a new type of user-controlled permission management scheme on the chain. This solution implements permission inheritance and permission grants by extending the CP-ABE model flexibly in enterprise-level scenarios. This scheme uses a multi-tree structure to implement the specification description of the attribute set, and on this basis, realizes the inheritance of permissions based on role assignment. The main tasks completed in this paper are as followed.

Aiming at the problem that CP-ABE increases in complexity as the number of users increases in enterprise-level application scenarios, this paper proposes an extended CP-ABE model based on role assignment. This model introduces the concept of attribute tree based on CP-ABE. The attribute tree satisfies the strict partial order relationship and can realize data user role division and authority inheritance operations. The system can construct an attribute tree according to the authority-contained relationship between roles. Each node in the tree except the root node and leaf nodes represents the role, and the leaf nodes in the tree represent each role's inherent attributes. If there is a path between two roles in the attribute tree, the owners of the two roles' permissions have an inheritance relationship. When CP-ABE performs data encryption, it constructs an access control structure based on the reachable relationship between roles in the attribute tree and the roles' inherent attributes to implement authority inheritance and implement hierarchical authority management based on the hierarchical relationship between roles. Based on the above scheme, the realization and verification of the prototype system of rights management on the chain were completed, and its performance was tested and analyzed. The prototype system can realize the independent and controllable rights distribution and control of the Blockchain data owner—permission inheritance function.

2 THE ACCESS CONTROL TREE

2.1 Access Control Tree Architecture

In the CP-ABE encryption system, the denater encrypts the data with a specific access control structure, determining which attributes the user can correctly decrypt. The denater realizes fine-grained access control by using the access control structure. The direct access structures are gate access structure, LSSS matrix access structure, and access control tree. The access control tree can flexibly represent the complex access structure and is more intuitive than the other two access structures.

Use T to represent the access tree, and each non-leaf node in the tree represents a threshold with child nodes and a threshold value. We use num to represent the node x number of child nodes, k is

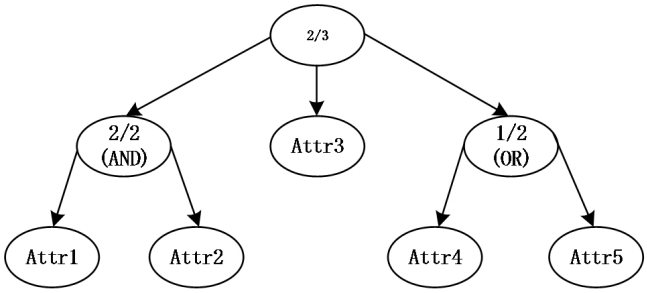


Figure 1: The Access Control Tree

the node threshold, where 0 < k ≤ num. When k = 1, the threshold represents or gate, and when k = num, the threshold represents and gate. Each leaf node of the tree represents an attribute, and the threshold value of the leaf node is 1. The access control tree structure can intuitively let users know what property groups can decrypt the corresponding ciphertext. An instance of the access control tree is shown in Figure 1

All leaf nodes are related to an attribute, all internal nodes represent a threshold, and 2/3 means that the threshold has three child nodes, and the threshold value is 2. That is, the user attribute needs to satisfy any two of the three child nodes. 2/2 means that the internal node has two child nodes, and the threshold value is 2. To satisfy the node, the user needs to satisfy both attribute 1 and attribute 2. That is, the node is equivalent to an and gate. 1/2 indicates that the internal node has two child nodes, and the threshold value is 1. To satisfy this node, the user needs to satisfy attribute 4 or attribute 5. That is, the node is equivalent to or gate. The access control tree structure in Figure 1 shows that if the user attribute set satisfies (attribute 1 and attribute 2) and attribute 3 or (attribute 1 and attribute 2) and (attribute 4 or attribute 5) or attribute 3 and (attribute 4 or attribute 5), the user attribute set satisfies the access control tree structure. Furthermore, the ciphertext can be decrypted correctly.

2.2 CP-ABE Algorithm Process

The CP-ABE algorithm process mainly includes four phases: parameter initialization, data encryption, key generation, and data decryption.

In the parameter initialization phases, the initialization algorithm's input has only one hidden security parameter, and there are no other parameters. The output of the initialization algorithm is the public key PK and the master key MK.

In the key generation phase, the key generation algorithm's input is the master key MK and a set of attributes describing the key associated with the user who wants to get the message M. The output of the key generation algorithm is the private key SK.

In the data encryption phase, the data encryption algorithm's input is public key PK, message M, and access control structure A, covering specific attributes. The algorithm will encrypt message M and generate ciphertext CT. Only those who have a set of attributes that meet the access control structure can correctly decrypt the ciphertext. Therefore, it can be considered that the ciphertext contains access control structure A.

In the data decryption phase, the data decryption algorithm input is public key PK and ciphertext CT. The ciphertext contains access control structure A and private key SK, which is generated according to a group of attributes S of users. If a set of attributes s satisfies the access control structure A, the user can decrypt the ciphertext and return the message M correctly.

3 THE ARCHITECTURE OF ACCESS CONTROL

3.1 Abbreviations and Acronyms

This paper proposes a new type of user-controlled permission management scheme on the chain. This scheme expands the access control tree in the CP-ABE model and proposes the concept of attribute tree. At the same time, blockchain technology is used to realize permission inheritance and permission grants. Compared with the traditional access control scheme that uses the CP-ABE model, this article's scheme removes redundant authorization and significantly improves authorization efficiency. The data generation architecture of the rights management scheme of this article is shown in Figure 2

The data owner owns the original private data D, and the data generation process is as follows:

- The original private data D is divided into data blocks, and the data owner determines the block standard according to its data access control requirements. After the data is divided into blocks, the data set D={D1, D2,..., Dn} is obtained.
- Use the symmetric key set K={K1, K2,..., Kn} to encrypt the data set D separately, set the encryption algorithm to f, and encrypt the ciphertext set C={C1, C2,..., Cn}, with Ci =f (Di, Ki). The ciphertext set C will be stored in the cloud database.
- To ensure the integrity of the private data stored in the cloud database, the symmetric key set K is used to calculate the message digest set H={H1, H2,..., Hn} for the data set D through the HMAC algorithm, where Hi=HMAC(Di, Ki). Store the message digest set H on the Blockchain.
- Use the improved CP-ABE algorithm to encrypt the key set K, and get the ciphertext set CTK={CTK1, CTK2,...CTKn} of the key set K, where CTKi=CP-ABE(Ki), encrypt the key The anthology CTK is stored on the Blockchain.

Finally, the metadata data format stored on the Blockchain is {data block description: dataNotei, key ciphertext: CTKi, HMAC value: Hi}. Data is not the data block description added by the data

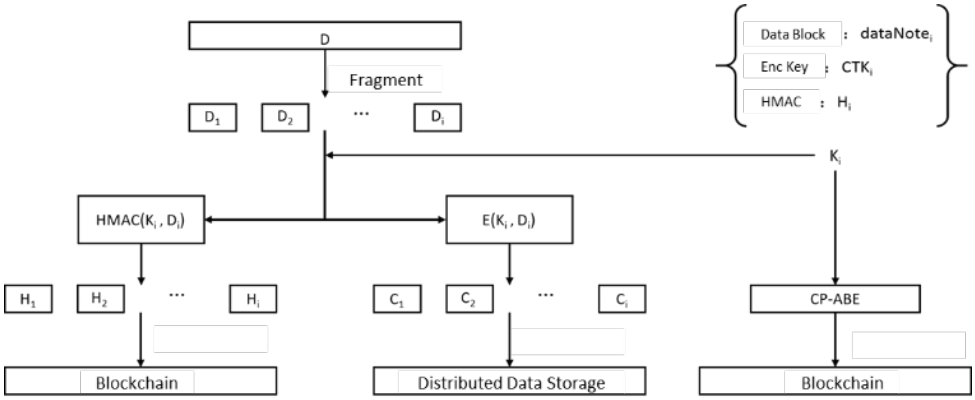


Figure 2: The architecture of distributed data storage with Blockchain

Table 1: Node attribute

Node	Directly attribute	Valid attribute
A0		{1,2,3,4,5,6,7}
A1	{1}	{1,3,4,6,7}
A2	{2}	{2,5}
A3	{3}	{3,6,7}
A4	{4}	{4}
A5	{5}	{5}
A6	{6}	{6}
A7	{7}	{7}

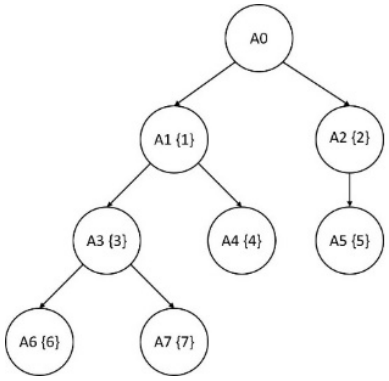


Figure 3: The attribute tree

owner for the privacy data block Di. The encrypted private data ciphertext set C will be persistently stored on the cloud database.

3.2 Construction of Attribute Tree

The attribute tree proposed in this paper is a tree structure containing roles and role attributes. In the attribute tree, leaf nodes represent role attributes, non-leaf nodes and non-root nodes represent roles. The inheritance relationship of roles in the attribute tree represents the inheritance relationship of permissions between users. Since role attributes do not directly control the inheritance of permissions, role attributes will not be considered in the following discussion. The nodes in the attribute tree except the root node represent roles, and the concepts of roles and attributes are no longer distinguished and used Property description uniformly. The attribute tree is shown in Figure 3. The attribute tree proposed in this paper uses the parent-child node relationship to represent the inheritance relationship of the attribute. The parent node contains all the attributes of the child node.

The attributes contained in all the nodes in Figure 3 are shown in Table 1

Nodes A4, A5, A6, and A7, have their own attributes 4, 5, 6, and 7, respectively. The child nodes of node A3 are A6 and A7. A3 inherits A6 and A7's attributes, and the valid attributes of A3 are 3, 6, 7. The effective attributes of A2 are 2, 5. A1 inherits the attributes of A3, A4. A3 inherits the attributes of A6, A7. So A1 also indirectly inherits A6, A7's attributes, and its valid attributes are 1, 3, 4, 6, 7. The root node A0 contains all attributes.

In addition to the direct inheritance relationship that can be represented by parent-child nodes in the property tree, there may also be indirect inheritance relationships. This inheritance relationship cannot be directly obtained through the parent-child node structure in the tree. Therefore, this article uses the reachable matrix to represent the indirect inheritance relationship. The element represents whether there is an inheritance relationship between the two attributes. Let $T=<V,E>$ be an attribute tree with N nodes

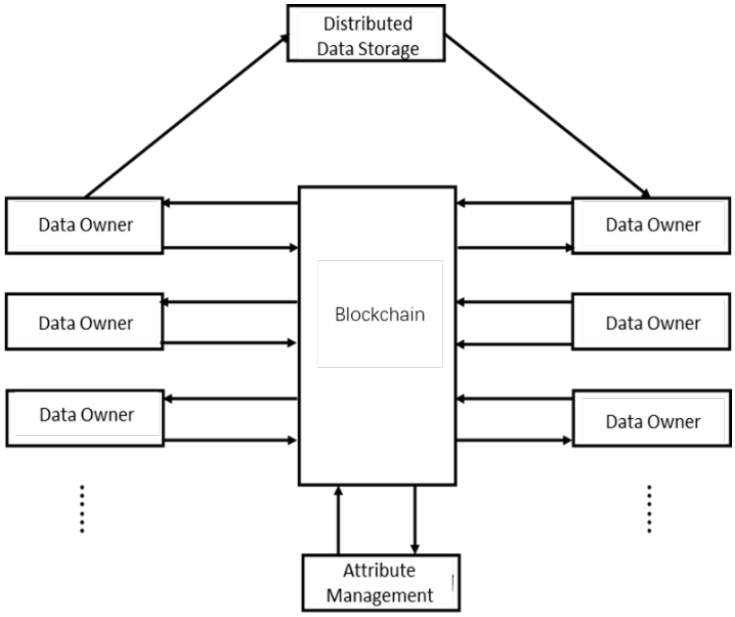


Figure 4: The system architecture

$V=\{v1,v2,...vn\}$. The elements in the reachable matrix $P=(p_{ij})$ corresponding to the attribute tree T are defined as:

$$p_{ij} = \begin{cases} 1 & there\ is\ a\ path\ between\ v_i\ and\ v_j \\ 0 & otherwise \end{cases} \quad (1)$$

The existence of a path between v_i and v_j indicates that v_i inherits the attributes of v_j, and includes direct inheritance and indirect inheritance. Otherwise, it means that there is no attribute inheritance relationship between the two. Only the reachability matrix is required to show the valid attributes of all nodes in the attribute tree concisely and clearly. The reachable matrix of the attribute tree can be obtained by the adjacency matrix of the attribute tree. Still taking the above attribute tree T as an example, the elements in the adjacency matrix $A=(a_{ij})$ are defined as:

$$a_{ij} = \begin{cases} 1 & v_i\ and\ v_j\ are\ adjacent \\ 0 & otherwise \end{cases} \quad (2)$$

The adjacency between v_i and v_j means that v_i and v_j are parent-child nodes and are directly connected in the attribute tree structure. The algorithm for finding the reachable matrix P using the adjacency matrix A is as follows:

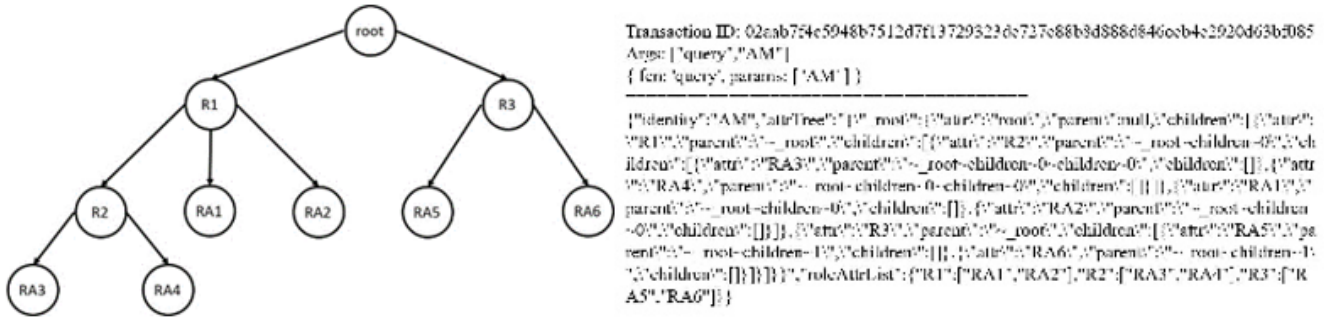
- Add the adjacency matrix A and the identity matrix E to obtain the matrix B, $B=A+E$.
- Perform the matrix operation of $B^2=B\times B$ on matrix B to obtain a new matrix $B=B^2=B\times B$.
- Repeat step 2 until the elements of matrix B do not change.
- Output reachable matrix $P=B$.

Take the attribute tree in Figure 4 as an example. Remove the node $A0$, and leave the adjacency matrix composed of nodes $A1$ to $A7$ as A. According to the above algorithm, the matrix B obtained from step 2 for the first time is $B=(A+E)^2$, The element value of the matrix obtained through step 2 for the second time will no longer change with the operation of step 2, so the matrix is the reachable matrix $P=B^2$ of the attribute tree.

After getting the reachability matrix, the inheritance relationship between attributes can be easily obtained, including direct inheritance and indirect inheritance. For a specific attribute v_i, it only needs to traverse the element p_{ij} in the matrix's corresponding row to check whether p_{ij} is 1. If it is 1, it means that the attribute v_i inherits the attribute v_j.

4 DISTRIBUTED DATA STORAGE MANAGEMENT ON BLOCKCHAIN

The architecture of the on-chain authority management system is shown in Figure 4. The system participants have three parties: data owners, data users, and attribute management agencies. The data owner stores his private data in the cloud database after symmetric encryption. Simultaneously, the symmetric key is encrypted by CP-ABE to generate the key ciphertext and stored in the Blockchain. The data user applies to the attribute management agency for roles and attributes through the Blockchain. The attribute management agency assigns the roles and attributes to the Blockchain data based on the data user's identity and credentials. According to the obtained role and attributes, the data user requests a key from the data owner through the Blockchain. The data owner uses the data user's role and attributes to generate a corresponding decryption key and distribute it to the data used through the Blockchain. The data user can obtain the key ciphertext through the Blockchain and decrypt it with its decryption key. If it has the authority, it will decrypt it correctly and then obtain the data owner's private data from the cloud database. If it is not assigned permissions, it cannot be decrypted and cannot obtain the data owner's private data. The process of authority management for the entire system mainly



(a) Attribute tree architecture

(b) Attribute management status data

Figure 5: Attribute tree and its storage format

includes the following steps: data user role assignment, authority chain release, and chain data acquisition.

The data owner first encrypts his private data in blocks to realize their private data's authority management. The encryption method adopts symmetric encryption. In the implementation of this chapter, the symmetric encryption algorithm uses AES-128. Simultaneously, the symmetric key is used to generate the data HMAC value, and the MD5 message-digest algorithm is used when generating the HMAC value. The encrypted private data is ciphertext encoded and uploaded to the cloud database, and the generated HMAC value is uploaded to the chain. After the data user obtains the private data, the HMAC value can determine the data's integrity. The symmetric key of the encrypted ciphertext data is encrypted by CP-ABE and on the chain. Different access control structures are specified in CP-ABE to realize access control. Suppose the data user specifies a role in the access policy. He needs to refer to the property tree maintained by the property management structure to find other roles inherited from the role and then use the CP-ABE extension model to encrypt the symmetric key, To realize the function of permission inheritance between roles. If the attribute tree structure is shown in Fig. 5(a), its storage form in the attribute management agency's status data is shown in Fig. 5(b).

CP-ABE encrypts the symmetric key K to generate a ciphertext file. It is assumed that the access control structure used during encryption is R1 and A1. That is, the data user whose role is R1 and attribute is A1 can decrypt the ciphertext. When the data user uses this strategy to encrypt the ciphertext, since there is no role inheritance role R1 in the attribute tree, it can directly use this strategy to encrypt the symmetric key. Encode the encrypted file content and HMAC together on the chain. These two parts constitute the data list dataList stored in the Blockchain by the data owner, which can be used as metadata for obtaining the data owner's private data. New data items are mainly added to the dataList. The data items mainly have three parts: data description data note, HMAC value hmac is used to verify data integrity, and key encryption The text ct, which is the ciphertext after using the CP-ABE encrypt the symmetric key.

Data users can check whether they have access to relevant private data by initiating a data request transaction. The data user adds the requested access data block information in the askAccessList item. The data access status for the data block D1 of the data owner DO1 is REQUEST, requesting. After the data request transaction is completed, the requested data owner initiates the access authorization to grant the transaction. DU1 initiates a data request to DO1, DO1 checks its role and attributes, and if DU1 has the authority to obtain the corresponding data, DO1 initiates the authority grant transaction. The request access list of the status data of DU1 askAccessList, the access status of the data block D1 that requests access to DO1 is changed from REQUEST to AGREE, which means that DO1 has agreed to the data access request of DU1. After the transaction is completed, DU1 can obtain the corresponding metadata from the Blockchain. DU1 uses its decryption key to decrypt ct to obtain the symmetric key K. DU1 obtains the ciphertext generated after encrypting D1 with K from the cloud database DU1 uses K to decrypt the ciphertext to obtain the corresponding private data.

5 EXPERIMENT AND EVALUATION

This section mainly tests the system from two aspects of latency and throughput, where latency refers to the time from the initiation of a transaction to the time the blockchain system responds to the transaction, and throughput is defined as the completed transactions per second Number, the unit can be expressed by TPS (Transactions Per Second). The prototype implementation of the on-chain permission management system is completed by invoking smart contracts to initiate transactions. In this process, two types of functions are mainly involved: query and invoke. The query is used to read the current status of the data owner, data user, and attribute management organization in the current status database. It is not a transaction in the strict sense and does not require the consensus of each node. It is just a process of merely querying status data. The participating parties' status data will be changed, but the throughput can still express its performance. All other functions belong to invoke. When the data owner, data user, and attribute management organization call this type of function, a transaction will be initiated to change the participating parties' state data in the state database. The relationship between the delay and the number of concurrent queries is shown in Figure 6(a), where the unit of the delay in milliseconds. The relationship between throughput and the number of concurrent queries is shown in Figure 6(b), where the number of concurrent queries on the abscissa represents the

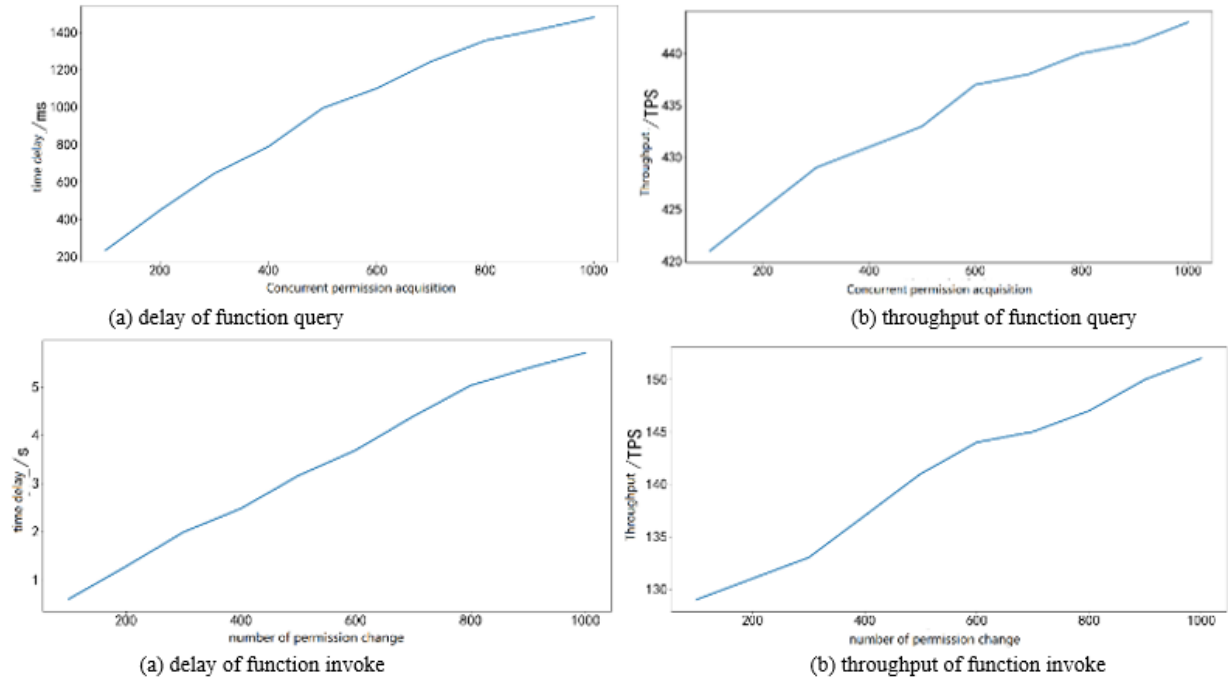


Figure 6: Performance of processing function

number of concurrent queries. It can be seen from the figure that as the number of concurrent queries increases, the delay will increase slower. At the same time, the throughput is also increasing, which can be maintained above 400TPS. It can be seen that the efficiency of calling a query to initiate a query is very high.

The process of initiating a transaction by all parties is the process of calling the invoke function. As the number of concurrent transactions increases, the transaction delay and throughput changes are shown in Figures 6(c) and 6(d), respectively. It can be seen that the delay of calling the invoke function is significantly higher than that of the query function. When the number of concurrent transactions is 1000, it is already about 5 seconds. Simultaneously, the throughput is significantly less than the query function, but it is also above 130TPS, and the efficiency is also high. Whether it is calling query or invoke, throughput is increasing. Here is a preliminary analysis because the blockchain system has not yet reached saturation, and there is still room for improvement. Although the throughput when initiating a transaction is lower than initiating a status data query, it also remains higher.

6 CONCLUSION

This paper proposes a new type of authority management model on the chain that users can control independently. The model combines the improved CP-ABE algorithm with the Blockchain to achieve fine-grained authority management while decentralization. Permission assignment and permission inheritance. The improved CP-ABE algorithm introduces a special attribute-role based on the traditional CP-ABE algorithm, representing the hierarchical relationship of the permissions owned by different data users in the organization. Further, the role of owner Role attributes, this hierarchical relationship can be refined through role attributes. To express this hierarchical relationship, the concept of an attribute tree is proposed. The attribute tree satisfies a strict partial order relationship. Its leaf nodes represent role attributes, and non-leaf nodes and non-root nodes represent roles. The existence of a path between roles represents an inheritance relationship between the permissions owned by the owners of these two roles. When data owners encrypt data, they will formulate access policies based on this inheritance relationship to achieve permission inheritance and construct data user permissions. The hierarchical relationship. The enterprise's inherent hierarchical relationship can be used to reduce its complexity in enterprise-level application scenarios through the improved CP-ABE model.

ACKNOWLEDGMENTS

This work is supported by the science and technology project of State Grid Corporation of China, "Research on Key Technologies of Blockchain Data Management for Trusted Sharing of Power Data" (Grand No. 5700-202072370A-0-0-00).

REFERENCES

[1] Bhattarai B P, Paudyal S, Luo Y, *et al.* "Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions". *IET Smart Grid*, vol. 2, no. 2. pp. 141-154. 2019

[2] Hossain E, Khan I, Un-Noor F, *et al.* "Application of big data and machine learning in smart grid, and associated security concerns: A review". *IEEE Access*, vol. 7, pp. 13960-13988. 2019

[3] Ghorbanian M, Dolatabadi S H, Siano P. "Big data issues in smart grids: A survey". *IEEE Systems Journal*, vol. 13, no. 4. pp. 4158-4168. 2019
[4] Wang K, Wang Y, Hu X, *et al.* "Wireless big data computing in smart grid". *IEEE Wireless Communications*, vol. 24, no. 2. pp. 58-64. 2017
[5] Chin W L, Li W, Chen H H. "Energy big data security threats in IoT-based smart grid communications". *IEEE Communications Magazine*, vol. 55, no. 10. pp. 70-75. 2017
[6] Li B, Kisacikoglu M C, Liu C, *et al.* "Big data analytics for electric vehicle integration in green smart cities". *IEEE Communications Magazine*, vol. 55, no. 11. pp. 19-25. 2017
[7] Hossain E, Khan I, Un-Noor F, *et al.* "Application of big data and machine learning in smart grid, and associated security concerns: A review". *IEEE Access*, vol. 7, pp. 13960-13988. 2019
[8] El-Hindi M, Binnig C, Arasu A, *et al.* "Blockchaindb: a shared database on blockchains". *Proceedings of the VLDB Endowment*, vol. 12, no. 11. pp. 1597-1609. 2019
[9] Wang S, Zhang Y, Zhang Y. "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems". *Ieee Access*, vol. 6, pp. 38437-38450. 2018
[10] Zyskind G, Nathan O, *et al.* "Decentralizing privacy: Using Blockchain to protect personal data". *2015 IEEE Security and Privacy Workshops. IEEE.* pp. 180-184. 2015
[11] Yao Y, Chang X, Music J, *et al.* "Lightweight and privacy-preserving id-as-a-service provisioning in vehicular cloud computing". *IEEE Transactions on Vehicular Technology*, 2019.
[12] Ouaddah A, Abou Elkalam A, Ait Ouahman A. "Fairaccess: a new blockchain-based access control framework for the internet of things". *Security and Communication Networks*, vol. 9, no. 18. pp. 5943-5964. 2016
[13] Ouaddah A, Elkalam A A, Ouahman A A. "Towards a novel privacy-preserving access control model based on blockchain technology in iot. Europe and MENA Cooperation Advances in Information and Communication Technologies". *Springer*, pp. 523-533. 2017
[14] Es-Samaali H, Outchakoucht A, Leroy J P. "A blockchain-based access control for big data". *International Journal of Computer Networks and Communications Security*, vol. 5, no. 7. pp. 137. 2017