Hindawi
Security and Communication Networks
Volume 2021, Article ID 3395537, 14 pages
https://doi.org/10.1155/2021/3395537

Research Article

Data Access Control Based on Blockchain in Medical Cyber Physical Systems

Fulong Chen, Jing Huang, Canlin Wang, Yuqing Tang, Cheng Huang, Dong Xie, Taochun Wang, and Chuanxin Zhao

Anhui Normal University, Wuhu, China

Correspondence should be addressed to Fulong Chen; long005@mail.ahnu.edu.cn

Received 16 July 2021; Revised 12 September 2021; Accepted 5 October 2021; Published 28 October 2021

Academic Editor: Zhe-Li Liu

The current medical cyber physical systems involve a wide range of institutions and a large number of participants. Data sharing among distributed medical institutions is already a development trend. However, the security is worrying; e.g., the access to medical data lacks uniformity and standardization. What is more, data is easy to be tampered with and leaked. This has a very negative impact on the medical industry. Therefore, a strict and reliable access control mechanism for data in the medical cyber physical systems is a prerequisite for ensuring the implementation of modern medical functions. We deal with how to design effective access control in medical cyber physical systems. Combined with blockchain technology, we design the medical cyber physical systems based on blockchain data access control mechanism and unite data in the chain of union Fabric network resources access control. We qualitatively classify medical data, define the weight level of different data, design a medical data access framework based on blockchain, build an applicable model, formulate access control strategy, and specify the role assignment and access task matching of users, so as to achieve secure and effective data access control. The Hyperledger Fabric network is established as the alliance chain for managing access control rights distribution through smart contracts so as to achieve case-based medical data access control under the blockchain.

1. Introduction

Medical cyber physical systems (MCPS) [1] are involved in various medical institutions, government health management departments, and other medical or health institutions and the important part of smart IoT healthcare [2]. Electronic health records (EHR) include structured data, semistructured data, and unstructured data. These massive data provide support for basic information transmission. Patients are the main data providers. Medical data is directly related to patients' privacy and life. Therefore, its importance is self-evident. To ensure the security of medical data and the privacy of patients is a necessary mechanism for MCPS. The mandatory implementation of access control for all users can stipulate that users can only access the specified resources, which is conducive to the standardized management of medical data access.

The larger the amount of medical data, the more obvious the clinical characteristics, the more extensive the sources, the more diverse the structure, and the more promoted the progress of medical services. Clinically, the data of patients with the same condition are different; e.g., the symptoms of patients with pneumonia are different in temperature, breathing, muscle contraction, sleepiness, and so on. In addition, in large tertiary hospitals, a large number of doctors rely on modern medical equipment to obtain and analyze data, while in community hospitals, many doctors still rely on patients' oral records, resulting in differences in medical data. Therefore, medical data itself is not easy to collect and store. If there is loss or malicious tampering, it will have a great impact on the diagnosis and treatment process. The types of medical data include structured data expressed in two-dimensional tables and logic, semi-structured data difficult to store and transfer, and

unstructured data with variable fields. The data must be firmly under control. Only patients have the highest authority for their own data, all data access and increase must be implemented according to the consensus reached by the system, and access records must not be erased, so as to ensure data sharing and data privacy protection.

In 2018, there were 18 data leaks in the United States, involving more than 100,000 medical records. Among them, eight accidents even affected more than 500,000 medical records, and three accidents resulted in the disclosure of more than one million medical records. The attackers obtained a lot of public medical information. In the traditional medical database systems, patients do not participate in the management of their own medical information, so that patients may not know who uses their data and for what purpose. The administrator with the highest authority can modify the access records of medical data, resulting in the loss of credibility of the data and inability to determine whose data and when is leaked. The medical system formed in this way has no credibility, and the medical data has lost its practicability, which is also a huge blow to the medical and health system.

In the MCPS, access control is an important means to ensure the security of medical data. It can manage the user's rights, so that legitimate users can only access the data in the system according to their own permissions and prohibit unauthorized users to access the data, so as to ensure the safety of data and the normal operation of medical systems. Nowadays, access control is still an important means to protect data transmission and sharing. However, as the management methods become more and more complex and the security requirements become higher and higher, the access control authorization management becomes more difficult, the description of access control objects becomes more difficult, and the tamper resistance of access control mechanism is poor. In this paper, blockchain is combined with access control. Visitors must be prevented from having too much medical data. Traceability and lossless modification of blockchain can improve the credibility of access control mechanism. Because it is difficult for the current public chain to meet the requirements of high throughput, low energy consumption, and privacy protection, the alliance chain with higher potential in performance, privacy protection, and permission control is suitable for the blockchain in the medical environment. The entities involved in the medical blockchain must be recognized by the government, have certain credibility, and be strictly supervised by the health management department. In this way, the occurrence of malicious behavior is far less than that of public chains such as bitcoin. At the same time, after years of medical information development, each hospital has a relatively complete network, server, and database system. Therefore, the existing medical information system can provide a relatively secure and stable operation environment for the normal operation of Practical Byzantine Fault Tolerance (PBFT) algorithm. At the same time, because each node in the cluster running PBFT algorithm has the same state, the medical blockchain system avoids the centralization of transaction block or blockchain and achieves distributed trustworthiness.

In this paper, the trusted mechanism of blockchain is used to design data access control in the MCPS. Section 2 will discuss related works. In Section 3, the medical data access framework based on blockchain is introduced. In Section 4, an access control strategy of medical data based on blockchain is proposed. The scheme is analyzed in Section 5.

2. Related Works

At present, data security and privacy protection technology in the medical environment is still constantly updated, especially in access control. Yu et al. [3] solved the problems of data confidentiality and fine-grained and extensibility of access control by defining and implementing access policies based on data attributes and allowed data owners to delegate most of the computing tasks involved in fine-grained data access control to untrusted cloud servers that do not disclose the basic data content. Wang et al. [4] proposed a data access control model for a single user. Through the semantic dependency between data and the bottom-up integration process, the global visibility of reverse XML structure is realized, which effectively protects the privacy and has high access efficiency. Zhu et al. [5] proposed a practical construction of attribute hierarchy-based encryption (ABE-AH) based on forward and backward derivation functions for cloud storage services by using composite sequential bilinear groups. This mechanism defines the priority of attributes, improves the granularity of data access control in cloud environment, and significantly reduces the size of key and ciphertext. Chen and Lin [6] proposed a new authorization access control model, which stores patient data according to the privacy level and obtains corresponding information according to different authorization modes. The privacy levels are set according to the specific situation. However, this model only solved the problem of access control of medical information for legitimate authorized users and did not involve other types of medical information leakage and security protection.

The access control studied by the above scholars all take into account the complexity of the medical environment and the difference of performance, which requires a third-party single-point entity or cloud server to make access control decisions, and the dependence on the third-party center is too high. Once the third-party center crashes, all nodes will be affected, and there is also the risk of data leakage. With the development of distributed computing, blockchain technology appears. Blockchain system uses chained block structure with time stamp to store data, so as to add time dimension to data. Each transaction on a block is password associated with two adjacent blocks, so that any transaction is traceable. Therefore, the combination of blockchain, a highly reliable technology, with the third-party center will greatly improve the reliability and performance and avoid the occurrence of single point of failure.

Cruz et al. [7] introduced a kind of role-based access control- (RBAC-) smart contract (RBAC-SC) using smart contract. The platform uses Ethereum's smart contract technology to realize cross organization role access. RBAC-SC uses smart contract and blockchain technology as a

common infrastructure to express the essential trust and recognition relationship in RBAC and implements a challenge response authentication protocol to verify the ownership of user roles. Xue et al. [8] proposed an electronic medical information sharing model based on blockchain technology, which helps to solve the problem of information sharing difficulty between medical institutions. Dubovitskaya et al. [9] proposed a secure and reliable electronic medical record system based on the traceability of blockchain, proposed a framework for managing and sharing EMR data for cancer patient care, and implemented the framework in the form of prototype platform to ensure privacy, security, and availability, as well as fine-grained access control of EMR data. Di Francesco Maesa et al. [10] used blockchain technology to define access control system to ensure the auditability of access control policy evaluation and wrote smart contract with eXtensible Access Control Markup Language (XACML) policy and solidness deployed on Ethereum blockchain. This idea is similar to that proposed in this paper, which shows the availability of smart contract to implement access control policy. Zhang et al. [11] aimed at the problems of inconvenient sharing of medical data, easy tampering, and easy leakage of privacy data, based on RBAC, used information entropy technology to quantify medical data, and used the distributed characteristics of blockchain and its inherent security attributes to eliminate data islands and enable patients to manage their own medical data independently. Tang et al. [12] proposed an electronic prescription sharing scheme based on blockchain and conditional proxy reencryption. The conditional proxy reencryption scheme can provide an efficient ciphertext forwarding mechanism for electronic prescription sharing. Ma et al. [13] proposed a decentralized access control model based on block chain smart contract implementation. After meeting the conditions set by the user, the user can apply for authentication to the blockchain to obtain the permission to access user data and operate user data to achieve secure access control of user data.

Through the above researches, we can see that the access control of medical data is still not perfect. At present, many scholars have been committed to combining the blockchain into the access control. However, the rights allocation, fine-grained access, and user management still need to be improved.

3. Medical Data Access Framework Based on Blockchain

Data is the basis of all applications in the MCPSs, and it is the most basic resource. Whether it is telemedicine, two-way referral, or artificial intelligence applications such as medical image detection and pattern recognition, a large amount of effective medical data are needed for testing. This work starts from the point that every access has a task. Combined with the concept of role and task, considering that access control can ensure the secure access after obtaining data, and blockchain can ensure the traceability of behavior, we design a medical data access framework based on blockchain, so as to improve the security of medical data protection.

3.1. Classification and Qualitative Analysis of Data. EHR mainly comes from patients. Therefore, patients have the highest authority of data, and they must know who can access their data, how to access it, and what to use it for. Other medical data come from medical institutions and medical devices themselves. The classification is shown in Table 1.

According to the sensitivity of medical data privacy protection, medical data is divided into four levels. The four types of data have different privacy sensitivity. The first type of privacy information has the highest sensitivity, and the corresponding weight should be the largest. The control of access rights should be stricter. The weight of the first, second, third, and fourth medical information decreases in turn, and the corresponding level gradually decreases.

Data 1: personal information of patients, including the basic information of patients (name, gender, date of birth, telephone number, ID number, home address, etc.), is the patient's personal privacy data, corresponding to the weight of L_1 which is used to represent the highest weight.

Data 2: patient's medical records, medical history, diagnosis and treatment records, and information collected by equipment, including patient's medical record data (disease type, diagnosis date, symptom description, image data, diagnosis conclusion, hospitalization records, diagnosis doctor, hospital information, etc.), are related to diagnosis and important medical data, which helps to analyze and conquer the disease, corresponding to the weight L_2.

Data 3: clinical outcome analysis and big data analysis are obtained through clinical and information technology, which can be used by a government department to analyze the condition and understand the national health, mainly reflected in the data integrity, corresponding to the weight L_3.

Data 4: medical institution information, equipment information, and other information are mainly about the history and qualifications of medical institutions, the storage capacity of equipment, and some publicly available data, which can be found in an open network environment, corresponding to the weight L_4.

In this paper, the medical data is divided according to the weight, which is conducive to fine-grained distinction, increases the flexibility of medical data, and provides a good foundation for privacy protection of medical data. The fine-grained structure of medical data adopts the structure combined with access weight, and the formal definition of medical data is described in the following formula:

Table 1: Symbols and definitions.

Class	Example
EMR or EHR	Electronic health records, diagnostic records Access records, patient personal information, etc.
Medical device data	Environmental data, network equipment data, medical device data, etc.
Information of medical institutions	Medical big data sorting and trend analysis, etc.

$$EHR = \begin{bmatrix} Data \\ L \end{bmatrix} = \begin{bmatrix} Data_1 & Data_2 & \dots & Data_n \\ l_1 & l_2 & \dots & l_n \end{bmatrix}. \quad (1)$$

3.2. Description of Subject and Object. In order to solve the problem of multiple users and complex tasks in the MCPS, the role-based access control model and task-based access control model are combined to classify users according to their roles. The access control authority is determined according to the task of the role, and the access authority, access policy, and access record are stored through the blockchain to ensure that the access authority cannot be tampered with, the access control can be made public, and the access record can be traced.

The unified description of user (U), subject (S), object (O), role (R), and task (T) in the process of access control can better express the access policy and be better combined with the access control model. In this paper, we define the following concepts:

(1) User: all kinds of people participating in the MCPS, including the participants of medical institutions, the staff of health management departments, and the staff of medical insurance related institutions. Each user has its own account.

(2) Subject: users who make access requests include their own ID information, resource information, and task information. In the medical environment, they will be bound with the corresponding roles.

(3) Object: the access resources, namely, medical data resources, are divided into $Data_1$, $Data_2$, $Data_3$, and $Data_4$ according to the privacy sensitivity of data.

(4) Role: in the medical environment, according to the characteristics of each institution, the roles that can represent the users of the institution are formed, such as patients, doctors, nurses, administrators, and technicians.

(5) Task: every time the subject visits the object, the purpose is carried out with the task of communication in the MCPS, and different users have different tasks to access data according to their own needs.

Different medical institutions have different departments. Although medical data is stored in the server of medical institutions, it is also classified and stored according to the department category. As shown in Figure 1, the general departments of Grade 3A hospitals are classified, and different roles belong to different departments. In the process of access, users cannot exceed the authority and can only access data according to the permission scope of the user's role. At the same time, when the subject accesses the data, the access control policy will also give the specific access path according to the specific access request information.

3.3. Data Access Control Model Based on Blockchain. Considering that each user in the MCPS has its own tasks, and a large number of users can be reasonably allocated by role classification, the role-based access control model and task-based access control model are combined, and the blockchain-based task role access control model (B-TRAC) is proposed to ensure the security of access control policies and access permissions by relying on the characteristics of trusted blockchain.

As shown in Figure 2, it is the schematic diagram of B-TRAC. The role-based access control model (RBAC) and task-based access control model (TBAC) are combined with blockchain. The combination of RBAC and TBAC lies in the data transfer between roles and tasks. As the requester of access, users are assigned roles in the session to reduce the difficulty of visitor management. According to the characteristics of MCPs, every visit carries a task to apply. Therefore, the tasks in the medical environment are classified and mapped with roles, so that the access can be fine-grained and specific access requirements can be made clear. The expected task information and access control permission information are stored in the blockchain formed by each organization in the medical consortium as a node. The blockchain can ensure that it will not be tampered with and leave traces every time. By matching the task information in the access request with the expected task information in the blockchain, it can be concluded that the task of the access request is quite the same as the expected task in the blockchain. If it is different, access will be denied. If it is the same, the blockchain will give the corresponding access permissions to the corresponding visitors to complete the settings before access.

3.4. Medical Data Access Control Architecture Based on Blockchain. Access to medical data is mainly for the personnel or medical equipment of medical institutions to collect data and send access requests. The framework of blockchain-based medical data access control (F-BMDAC) is shown in Figure 3. In order to ensure that the policy and permission exchange are publicly visible on the blockchain,

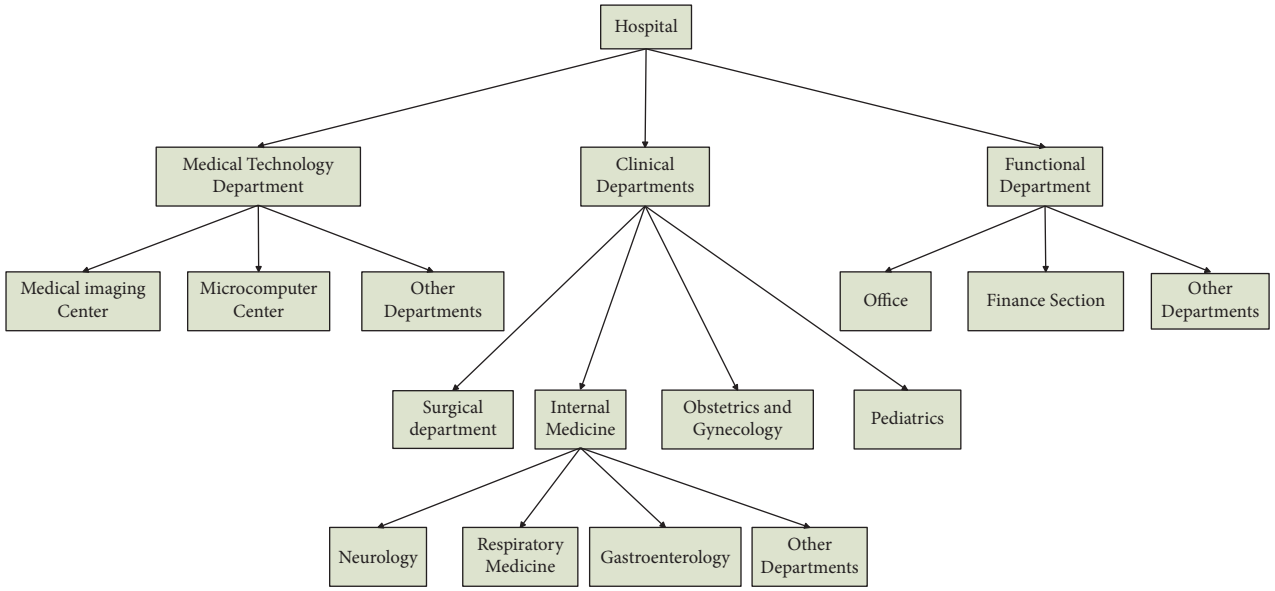
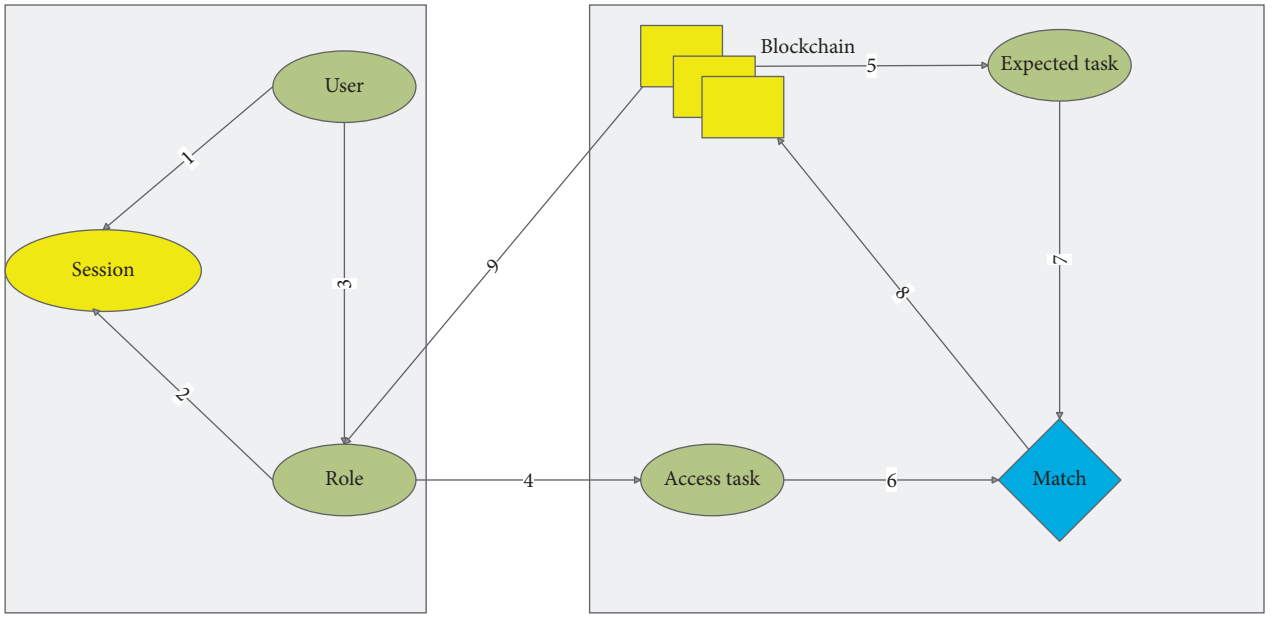


Figure 1: Department classification.



Directions:

1: The user applies for access
2: Role assignment, establish a session with the user
3: The user gets the role assignment information
4: The user releases access tasks with role information
5: Query the expected tasks in the blockchain
6, 7: Match the visit task with the expected task
8: The matching result is fed back to the blockchain
9: Blockchain grants corresponding access permissions to roles based on the feedback results

Figure 2: Blockchain-based task role access control model.

and to save blockchain resources, these two key data are stored on the blockchain. Other medical data are still stored on the server of the medical institution representing each node, and the server of the medical institution undertakes the work of identity authentication before access control.

After the user sends the access request, the data security management module will analyze the access request, assign roles, analyze tasks, and request the medical data access permission from the blockchain data storage module. The blockchain module evaluates and obtains the permission

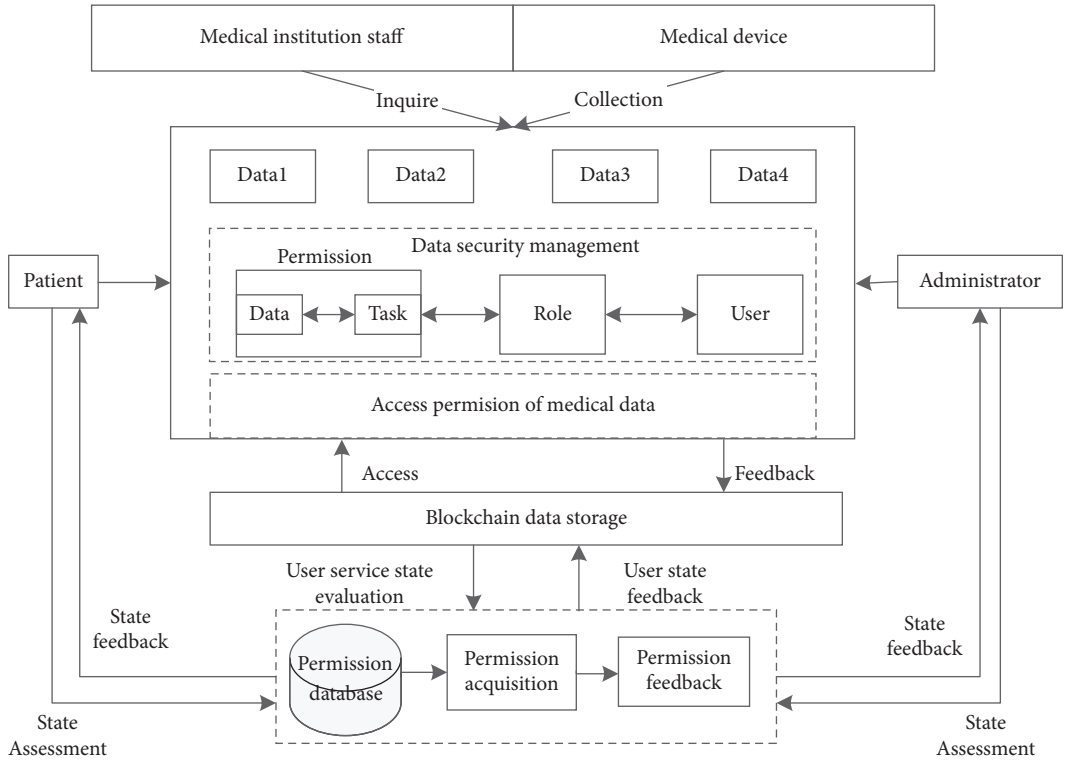


Figure 3: Framework of blockchain-based medical data access control.

database and feeds back the results. At the same time, the access to the permission information should also inform the patient to know the records of personal data being accessed. At the same time, the system administrator collects the access records and generates the access log.

4. Access Control Strategy of Medical Data Based on Blockchain

As the stage after identity authentication, access control is the means to determine the scope of data access. According to the proposed B-TRAC model and the F-BMDAC model, the data access control strategy based on blockchain in the MCPS is designed. The following describes in detail the role assignment, task assignment, task matching, and decision-making in the whole access control framework, as well as the access control permission management. Introducing the concept of role is beneficial to manage a large number of users, and introducing the concept of task is beneficial to refine each access request to achieve the purpose of fine-grained access control.

The medical data in the MCPS is mainly stored in the server of each medical institution. The highest decision maker is the patient himself whether it can be accessed or not. Considering the high frequency utilization rate of medical data, patients authorize blockchain nodes in advance to manage data, and patients can check the access records of data at any time. When the access request is to consult data, the visitor and the blockchain node can exchange information and authorize access, but the authorization records and access records must be stored on the blockchain to ensure transparency.

4.1. User Role Assignment. The Central Server (*CS*) of each medical institution audits the access request (*AR*) information according to the authenticated information of the user node (*UN*), obtains the initial identity of the server, and assigns roles to the user according to the role type classification rules, which is recorded as

$$\{UN \leftarrow R\}. \tag{2}$$

Step 1. *CS* audits *AR* information and obtains $UN = \{ID, Task\}$. According to the user's ID information, the role R is assigned to the user, represented as

$$UN' = \{ID, Task, R\}. \tag{3}$$

Step 2. If the *ID* in *AR* does not pass the registration in the authentication stage, the corresponding *ID* does not exist in *CS* and is not given the role, and the error information is recorded in the server log.

4.2. Task Information Matching. In order to enhance the practicability of MCPS, it is required that every access must carry the access task requirements; otherwise, it will not accept the access. At the same time, carrying access task is also for better decision-making, to distinguish the legitimacy of access. Access task (AC) is also determined according to the actual needs, as shown in Figure 4. In the MCPS, the expected task (*ET*) need to be defined in Enquiry, Medical, Research, Insurance, and other access categories. Each access task should be classified according to the category to prevent

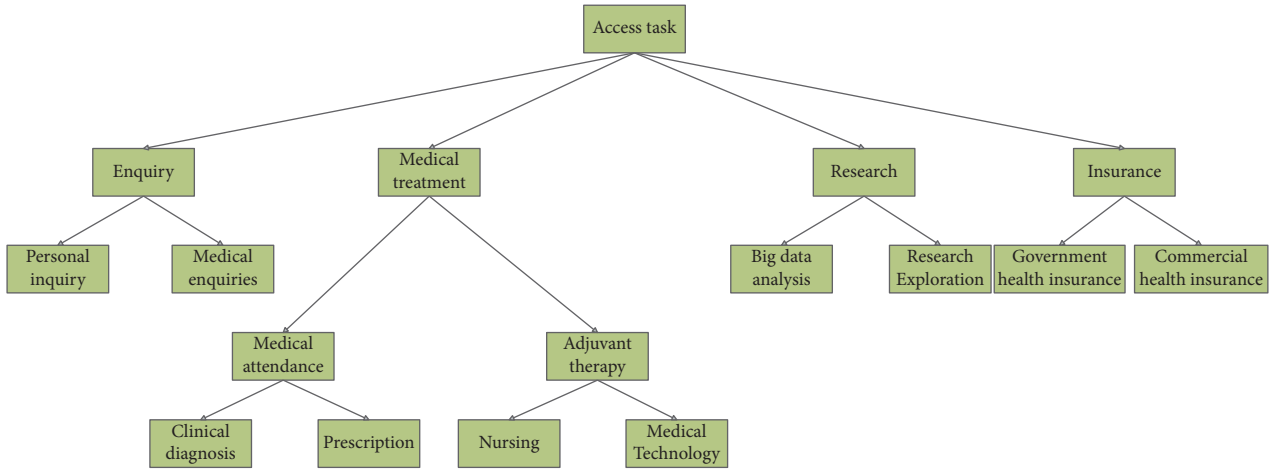


Figure 4: Expected access task tree.

malicious access and unauthorized access. Each access task is classified into the expected access task tree, which can ensure that each access application carries a legal access task. And at the same time, it can achieve fine-grained access requirements, make the specific category of each access clear, and form a mapping relationship between the access task and the user's role.

Whether the access control request can be allowed or not depends on whether the access task and the expected task match. The access task is carried by the access request, and the expected task is stored on the blockchain, which is the key to determining whether the access control request is allowed or not. The decision algorithm of access task matching is shown in Algorithm 1.

The main tasks of the algorithm are as follows: (1) put all the expected task sets waiting for matching into the task as the buffering task set and (2) traverse the buffering task set to get the matching result sets including permit and reject, respectively.

4.3. User Permissions Management. The access users of medical data may come from different medical institutions and play different roles. Therefore, it is necessary to authorize, verify, update, and revoke the access permissions of users, as shown in Figure 5. The specific process is as follows.

Authorization. According to the B-TRAC access control model, the blockchain platform grants the corresponding permission level of medical data access to different roles, noted as

$$\{R \leftarrow P | L_i \bullet n\} i = 1, 2, 3, 4, \tag{4}$$

and saves it in the log of the blockchain, where n is the update factor provided by the blockchain.

Verification of Permissions. The user role R carries the access task AT. Under the condition that the user's identity is legal, the server CS of the node responds to the medical data access according to the corresponding permissions of the user in the $\{R \leftarrow P\}$ permission level L.

Update of Permissions. According to the access control model, when the authority is expired or the task value of medical data is added, the blockchain provides a new update factor rn' in time to replace the previous rn value, calculates the new medical data access permission,

$$P' = P \bullet rn', \tag{5}$$

and authorizes it to the corresponding role R, recorded as

$$\{R \Leftarrow P'\}. \tag{6}$$

Revocation of Permissions. According to the need of access control, the blockchain sends permission revocation instruction $\{R \longrightarrow revoke\}$, that is, revoking the access permission corresponding to role R.

5. Security Analysis and Discussion

5.1. Characteristic

5.1.1. Principle of Minimum Privilege. It means to ensure that the user can obtain the minimum permission on the premise of ensuring the completion of the access request, so as to prevent the phenomenon of excessive authorization. Combining the concept of role with the concept of task can better manage users and refine the access of each user, effectively prevent users from malicious access, reduce the impact of illegal operation and false users on the MCPS, and minimize the loss in case of hacker attack.

5.1.2. Principle of Separation of Duties. Roles all correspond to the concept of specific application background, such as patients, doctors, nurses, and managers, in the medical environment. Access permissions can be divided according to specific categories, which makes the management of access control very flexible and simple. In order to prevent users from having more permissions, the blockchain will assign permissions to the roles of users according to specific tasks. And patients can regularly query which visitors and access records their data access rights belong to. At the

```
Input: access task
AT, expected task
ET
Output: decision permit/reject
(1) permit ← null;
(2) reject ← null;
(3) Task ← AT;
(4) for i = 1 to Task.length do
(5)     result ← Decision (Task [i], ET); //Permission
(6)     match
(7)     if result is permitted then
(8)         Task.delete (Task [i]);
(9)         permit.add (Task [i].PID);
(10)    end
(11)    else if result is rejected then
(12)        Task.delete (Task [i]);
(13)        reject.add (Task [i].PID);
(14)    end
(15) end
(16) return permit, reject;
```

Algorithm 1: The decision algorithm of access task matching.

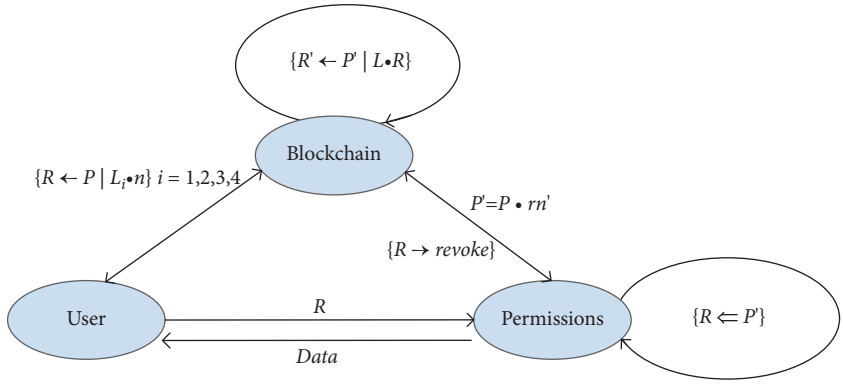


Figure 5: Authority management of medical data.

same time, this way can be fully applied in medical reimbursement to prevent the phenomenon of repeated reimbursement.

5.1.3. Tamper Proof of Access Permissions and Policies. The specific permissions and policies of access control are stored on the blockchain and can be queried by all nodes on the blockchain, which shows the openness and transparency of access control. After completing the task matching, the blockchain node is authorized. Due to the credibility of the blockchain, the access permissions and policies cannot be tampered with.

5.1.4. Fine-Grained Access Control. Patients have the highest control permissions over medical data. They can clearly specify the authorization and revocation of data permissions, including the time and frequency of sharing and the specific scope. Only authorized users are allowed to access the data specified by patients. Users are abstracted and classified by roles. Compared with a large number of users, roles can be more convenient for system management and reduce the burden of system administrators. By specific-to-each-visit task, in the medical task tree and fine-grained access control, we can avoid that each visit is not specific.

5.2. Security Analysis

5.2.1. Security 1: Confidentiality of Medical Data. The patient data is stored in the server of the local medical institution, which cannot be accessed by unauthorized users. The data stored on the blockchain through hash operation and Merkle tree structure are digests. This kind of on-chain and off-chain storage structure can reduce the performance pressure of blockchain and avoid the direct disclosure of source data on the network.

5.2.2. Security 2: Tamper-Resistant Medical Data. Each data block in the blockchain stores the hash value of its parent block, which is arranged in a certain sequence. The Practical

Byzantine Fault-Tolerant (PBFT) algorithm can be used as a consensus algorithm in medical blockchain. PBFT algorithm does not need as much computing power as PoW algorithm to avoid "51% attack." As a Byzantine Fault-Tolerant (BFT) algorithm, pbft algorithm has errors or malicious nodes less than or equal to those in the system to ensure the normal execution of distributed consensus process and the data cannot be tampered with.

5.2.3. Effective Access and Sharing of Medical Data. The permission of data sharing is entirely determined by the data provider. Based on the principle of who provides data and who has the highest authority, the decision algorithm of access control policy must be satisfied in every access control decision.

5.2.4. Single-Point Attack Risk. All information on the medical alliance blockchain is open and tamperable. The ledger in hash data is stored in the form of copies on each node in the network. In this way, the decentralized distributed structure does not have the problem of single-point attack in the traditional centralized organization.

5.2.5. Anti-DDoS Attack. Distributed Denial of Service (DDoS) attack is a common problem in distributed system architecture. MCPS involves multiple domains and has the distributed features. The blockchain platform needs high-performance server as support, which ensures that the device is not a bottleneck. At the same time, in order to prevent a large number of useless users, the blockchain data not only limits the validity of permissions but also sends its authorization records to the data provider, so that it can resist DDoS attacks to a certain extent.

5.3. Security Proof of Blockchain. The security threat of blockchain mainly comes from the attacker's attack on block and consensus mechanism, so as to achieve the purpose of modifying block data. We define p as the probability that the trusted node calculates and generates the next block, q as the probability that the malicious node of the attacker calculates and generates the next block, and q_n as the probability that the attacker calculates n nodes to complete the attack, as shown in the following formula:

$$q_n = \begin{cases} 1 & p \le q \\ \left(\frac{q}{p}\right)^n & p > q \end{cases} \tag{7}$$

In the case of $p > q$, assuming that it takes an average time for the trusted node to calculate a block data, the length of the attacker's forged blockchain will conform to Poisson distribution, and the mathematical expectation λ is shown in the following formula:

$$\lambda = n \times \frac{q}{p}. \tag{8}$$

The Poisson distribution probability density function of new quantity that attackers forge new blocks of the blockchain multiplies the probability that the attacker successfully chases the trusted blockchain under this number, that is, the probability of the attacker successfully tampering with the block data P is shown in formula [8]:

$$P = \sum_{k=0}^{\infty} \frac{\lambda^k \cdot e^{-\lambda}}{k!} \cdot \begin{cases} \left(\frac{q}{p}\right)^{(n-k)}, & k \le n \\ 1, & k > n. \end{cases} = 1 - \sum_{k=0}^{n} \frac{\lambda^k \cdot e^{-\lambda}}{k!} \cdot \left(1 - \left(\frac{q}{p}\right)^{(n-k)}\right), \tag{9}$$

It is concluded that the attacker must obtain the function of 50% nodes in the blockchain network in order to control the whole blockchain data.

6. Design and Verification of Hyperledger Prototype Platform

6.1. Experimental Environment. Access control mechanism is built on the basis of a more perfect identity authentication mechanism. In this section, through the Fabric 1.4 version of Hyperledger, with its membership module as the basis of identity management, and through the chain code services module, the access control mechanism is implemented with smart contract to complete the user (visitor) access (transaction) request to the ledger data. Considering the integrity of medical information physical fusion system in the actual operation, the data in the medical environment is described as a unified object. In order to take into account the heterogeneity of medical data and reduce the cost and complexity of medical informatization, the concept of attribute is used in the experiment to manage it uniformly. The premise of access control is identity authentication, and the foundation is certificate mechanism. There are three types of certificates in Fabric, such as Enrollment Certificate (ECert), Transaction Certificate (TCert), and TLS Certificate (TLSCert) used to guarantee the transmission security of communication link. The default signature algorithm of certificate is ECDSA, and the hash algorithm is SHA-256. ECert is issued to the principal who has provided the registration certificate to represent the identity of the principal in the Fabric. TCert is issued to users to control the permissions of each access. Each access can be

different, so as to achieve anonymity. TLSCert controls the access of the network layer, which can verify the identity of the remote subject and prevent eavesdropping. In practical application, the identity of the main body is verified by ECert, and the authority management is achieved by checking the signature. For the administrator role in the blockchain, other operations depend on the definition in the MSP structure of the corresponding organization in the channel configuration, except that the installation chain code operation is to find and match the certificate list under the msp/admincerts path of the signed certificate root node.

6.2. Attribute Definition and Mapping of Object. Fabric network can restrict the ability of access nodes and users through PKI-based membership management. The decision algorithm of access control can be realized by the chain code of identity attribute when Hyperledger Fabric is running. In order to complete the extraction of attributes, the registration certificate of identity is required in membership. ECert contains one or more attribute names and corresponding attribute values as shown in Table 2.

6.3. Data Structure. In fact, blockchain connects all blocks through chain structure, which can be divided into internal block structure and inter block structure. Block is the node unit of the blockchain, and Genesis block is the starting node. The blockchain node is divided into block head and block body. After Genesis block, the block head of each blockchain contains the parent block hash, version number, consensus metadata, timestamp, status hash, Merkle root, and other information. Mining blockchain applications such as bitcoin also have the target difficulty value to control the mining difficulty. Block body contains block format, block size, transaction details or summary, and other information. As shown in Figure 6, the blockchain node in Hyperledger Fabric is taken as a case to show the data structure of the blockchain. The application of blockchain is different, and the internal information is slightly different.

Merkle tree is applied in blockchain mainly for two reasons.

6.3.1. PV-Simplified Payment Verification. If someone needs to verify whether a transaction information exists in the blockchain, he or she only needs to obtain the block header in the blockchain node and the complete Merkle tree where the transaction needs to be verified; that is, to perform a SPV proof: get the hash value of the transaction from the node, locate the block, download its complete Merkle tree, recalculate, and verify whether the Merkle root value is equal to the block head; that is, the verification is passed. Of course, SPV verification does not need to calculate all the hash values. It only needs to calculate the value on the binary tree path where the transaction information is located to complete the verification quickly. For the block containing n transactions, the complexity of SPV verification is $\log_2 n$, which can ensure the integrity of the data.

Table 2: Attribute information mapping.

Attribute name	Attribute value
attrID	ID of the attribute
attrName	Name of the attribute
AttrValue	Value of the attribute

6.3.2. The Cost of Forging Merkle Roots Is Too High. Because of the irreversibility of hash operation, each node of the binary tree hashed from the bottom of the network is closely linked. If the data on the blockchain is maliciously damaged, the Merkle root and block header hash values will change, resulting in the block header hash change in the next block, thus causing the change of the nodes in the whole network. Therefore, if hackers want to attack blockchain nodes, they must control more than 51% of the computing power of the nodes. In the current complex large-scale heterogeneous network environment and the background of powerful data processing capacity, this situation is almost impossible to achieve, which ensures that the data can not be tampered with.

6.4. Management of Access Permissions. In the MCPS, different permissions of different users need to be set, which can be executed through the chain code in Fabric. In this paper, the user tag attribute is used to realize, and the chain code call is used to verify the permissions.

User registration: when the SDK applies for a user, it specifies the user tag and obtains the corresponding tag when the chain code is executed to realize the chain code call permission verification. MSP stores copies of three types of certificates: administrator certificate admincerts, root certificate cacerts, and TLS root certificate tlscacerts. However, only the certificate is stored, and the private key is not stored. CA stores the root certificate of a functional certificate and its corresponding private key. The specific method of user registration in this paper is shown in Algorithm 2.

Algorithm 2 deals with the following functions:

(1) Taking the identity attribute, attribute name, and password as test cases

(2) Setting corresponding values for attribute information

(3) Testing the correctness of the password and outputting the ID in MSP.

Chain code operation: the access initiator will belong to the corresponding organization org in the Fabric. The cryptogen tools provide certificates for different organizations and organize these certificates into the forms that can be directly used by the core components of peer and orderer. The chain code operation method of authority management in this paper is shown in Algorithm 3.

The main tasks of Algorithm 3 are as follows:

(1) Getting the initiation information of visitors

(2) Reading the corresponding org, peer, and user values

(3) Sending feedback read log

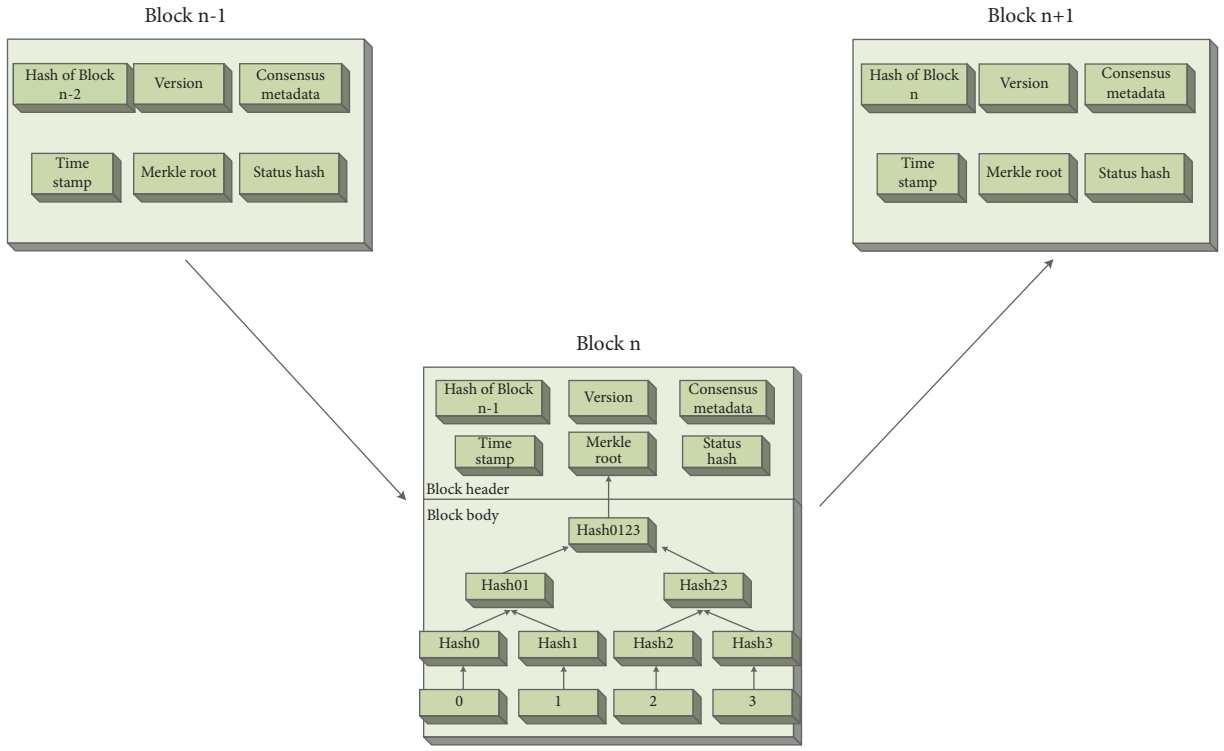


Figure 6: Blockchain data structure.

```
    Input: attrID, attrName, pwd
    Output: user.setMspId which is the ID of the user in MSP
(1) FabricUser user ← getUser (attrID, attrName, pwd);
(2) RegistrationRequest request ←newRegistrationRequest (attrID, attrName, pwd);
(3) request.setSecret (pwd);
(4) request.addAttribute (new Attribute ("attrID "));
(5) request.addAttribute (new Attribute ("attrName "));
(6) request.addAttribute (new Attribute ("pwd "));
(7) RegistrationRequest request = new
    RegistrationRequest (attribute);
(8) request.setSecret (pwd);
(9) if !user.getEnrollmentSecret ().equals(pwd) then
(10)    throw new RuntimeException (" The password is
(11)    abnormal. The password you set is inconsistent
(12)    with the password in the system: yourPWD:"
(13)    +pwd+," system: "+user.getEnrollmentSecret ());
(14) end
(15) user.setEnrollment (ca.enroll (NAME, pwd));
```

Algorithm 2: User registration.

6.5. Experimental Analysis. Through the access to the case data in Fabric, set the user's access permissions to different data, manage and classify users through attributes, and distinguish the permissions of the logged in user's attribute information, and the experimental verification system achieves the effect of access control. The CA in MSP is

```
Input: the information of the access initiator, such as the organization org, the node peer, and the user ID
Output: the access log file of the subject and the operation record of the chain code
(1) logger.Info (Sprintf ("get args: %s," args)); //Get the information of the access initiator
(2) / ∗ read org ∗ /
(3) if err ! = nil then
(4) logger.Error (fmt.Sprintf ("get deptAttrVal err: %s," err.Error ()))
(5) end
(6) else
(7) if deptAttrVal = = df then
(8) logger.Info (fmt.Sprintf ("get deptAttrVal: %s," dv));
(9) end
(10) else
(11) logger.Debug (fmt.Sprintf ("not found deptAttr"));
(12) end
(13) end
(14) / ∗ read peer ∗ /
(15) if err ! = nil then
(16) logger.Error (fmt.Sprintf ("get orgAttrVal err:%s," err.Error ()));
(17) end
(18) else
(19) if orgAttrVal = = of then
(20) logger.Info (fmt.Sprintf ("got orgAttrVal: %s," ov));
(21) end
(22) else
(23) logger.Debug (fmt.Sprintf ("not found orgAttr"));
(24) end
(25) end
(26) pv, pf, err \coloneq sinfo.GetAttributeValue ("peer");
(27) / ∗ read user ∗ /
(28) if err ! = nil then
(29) logger.Error (fmt.Sprintf ("get peerAttrVal err: %s," err.Error ()));
(30) end
(31) else
(32) if peerAttrVal = = pf then
(33) logger.Info (fmt.Sprintf ("got peerAttrVal: %s," pv));
(34) end
(35) else
(36) logger.Debug (fmt.Sprintf ("not found peerAttr"));
(37) end
(38) end
(39) if err! = nil then
(40) logger.Error(fmt.Sprintf ("get userAttrVal err: %s," err.Error ()));
(41) end
(42) else
(43) if userAttrVal = = uf then
(44) logger.Info (fmt.Sprintf ("got userAttrVal: %s," uv));
(45) end
(46) else
(47) logger.Debug (fmt.Sprintf ("not found userAttr"));
(48) end
(49) end
(50) return shim.Success ( [] byte ("log"));
```

Algorithm 3: Permission management.

```
root@fabric-cli:/tmp# peer chaincode instantiate \
> -o orderer.example.com:7050 \
> -C "businesschannel"  \
> -n test02 \
> -v 1.0 \
> -c '{"Args":["init","a","100","b","200"]}' \
> -P "OR('Org1MSP.member','Org2MSP.member')" \
> --tls true \
> --cafile /etc/hyperledger/fabric/crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
[001 03-08 09:40:21.01 UTC] [github.com/hyperledger/fabric/msp/mgmt] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.InitCmdFactory.GetDefaultSigner.GetLocalMSP -> DEBU Returning existing local MSP
[002 03-08 09:40:21.02 UTC] [github.com/hyperledger/fabric/msp] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.InitCmdFactory.GetDefaultSigner.GetDefaultSigningIdentity -> DEBU Obtaining default signing identity
[003 03-08 09:40:21.02 UTC] [github.com/hyperledger/fabric/peer/chaincode] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.instantiate.getChaincodeSpec.checkChaincodeCmdParams -> INFO Using default escc
[004 03-08 09:40:21.02 UTC] [github.com/hyperledger/fabric/peer/chaincode] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.instantiate.getChaincodeSpec.checkChaincodeCmdParams -> INFO Using default vscc
[005 03-08 09:40:21.03 UTC] [github.com/hyperledger/fabric/msp] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.instantiate.GetSignedProposal.Sign -> DEBU Sign: plaintext: 0A9A070A6C08031A0B08858A84D50510...324D53500A04657363630A0476736363
[006 03-08 09:40:21.03 UTC] [github.com/hyperledger/fabric/msp] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.instantiate.GetSignedProposal.Sign -> DEBU Sign: digest: F8F23AA8A4E9BD4A7B59119CDE27B43368256F049365F198A86EC95547E19CF0
[007 03-08 09:40:48.06 UTC] [github.com/hyperledger/fabric/msp] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.instantiate.CreateSignedTx.Sign -> DEBU Sign: plaintext: 0A9A070A6C08031A0B08858A84D50510...123DAF5C5FBF9407915DCE40521310C2
[008 03-08 09:40:48.06 UTC] [github.com/hyperledger/fabric/msp] main.Execute.ExecuteC.execute.func1.chaincodeDeploy.instantiate.CreateSignedTx.Sign -> DEBU Sign: digest: 59D67034C76950C4B07FA01CC51E197EB6D236DBE32A0AC9564679C0AAEBBBEA
[009 03-08 09:40:48.06 UTC] [main] main -> INFO Exiting.....
```

Figure 7: Query node information and signature value.

```
NAME                                DESCRIPTION                                     STARS               OFFICIAL
          AUTOMATED
sameersbn/gitlab                    Dockerized gitlab web server                    1144
          [OK]
sameersbn/redmine                                                                   311
          [OK]
sameersbn/squid                                                                     185
          [OK]
sameersbn/bind                                                                      174
          [OK]
sameersbn/postgresql                                                                151
          [OK]
sameersbn/redis                                                                     79
          [OK]
sameersbn/mysql                                                                     58
          [OK]
sameersbn/gitlab-ci                                                                 47
          [OK]
sameersbn/openfire                                                                  40
          [OK]
sameersbn/apt-cacher-ng                                                             37
          [OK]
sameersbn/gitlab-ci-multi-runner                                                    33
          [OK]
sameersbn/invoiceplane              Dockerfile to create a Docker container imag…   27
          [OK]
sameersbn/wowza                                                                     19
          [OK]
sameersbn/gitlab-ci-runner                                                          18
          [OK]
```

Figure 8: Data access.

responsible for the registration, management, and certificate issuance of users. org is used to represent a hospital in the MCPS, user is used to represent users, and peer is used to represent nodes other than users. As shown in Figure 7, after setting the information of the node pair, we can query the relevant specific information by using membership to manage it. At the same time, we can query the digital signature information to ensure the traceability of each visit.

In the decision of access control policy, the policy identifier must sign a specific user to satisfy the policy. The access control policy is managed by the chain code, and the permission information and the accessed data are stored on the chain. When the data is accessed successfully, it is shown in Figure 8.

7. Conclusions and Future Works

In this paper, we start with the data resources of MCPS and, combining with the blockchain technology, put forward the access control mechanism to ensure the secure access of data. The MCPS combined with blockchain is a general

trend, which can make use of the distributed and credibility of blockchain technology to disclose the records of each visit and ensure the secure storage of data [14]. The future work can be continued from two aspects. On the one hand, considering the complexity of medical environment and the difference of device performance, we can design lightweight access control mechanism to ensure access security in heterogeneous environment. On the other hand, we can set up a unified data access control mechanism to cooperate with various medical institutions to ensure the security of data. MCPS covers a wide range of institutions and involves people's life safety and property interests, which is related to the development of hospitals, research institutes, insurance companies, and government departments [15]. Therefore, the comprehensive consideration of how to meet the access control standards of different users in different demand conditions will promote the development of the whole medical informatization and better guarantee the security of medical data.

Data Availability

All the experiments were run in the virtual machine Ubuntu. As a result, in the paper, we did not collect the data separately. At that time, we used Fabric's own test data and then simulated the node data transmission. We confirm that the manuscript is not under review or published elsewhere.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank their colleagues and students in Anhui Provincial Key Laboratory of Network and Information Security. The authors thank National Natural Science Foundation of China, under Grant no. 61972438, and Key Research and Development Projects in Anhui Province, under Grant no. 202004a05020002 for supporting this research.

References

[1] F. Junior, D. Schneider, and R. Adler, "Dynamic risk management for cooperative autonomous medical cyber-physical systems," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, pp. 216–231, Turku, Finland, September 2019.

[2] Y. Zhang, Y. Sun, Y. Sun et al., "High-performance isolation computing technology for smart IoT healthcare in cloud environments," *IEEE Internet of Things Journal*, p. 1, 2021.

[3] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, San Diego, CA, USA, August 2010.

[4] M. Wang, J. Wang, L. Guo, and L. Harn, "Inverted XML access control model based on ontology semantic dependency," *Computers, Materials & Continua*, vol. 55, no. 3, pp. 465–482, 2018.

[5] Y. Zhu, D. Huang, C.-J. Hu, and X. Wang, "From RBAC to ABAC: constructing flexible data access control for cloud storage services," *IEEE Transactions on Services Computing*, vol. 8, no. 4, pp. 601–616, 2015.

[6] Y. Chen and Y. Lin, "Research on the method of authorized access controlling in medical information privacy protection," *Journal of Healthcare Information Management*, vol. 15, no. 3, pp. 288–291, 2018.

[7] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[8] T. Xue, Q. Fu, C. Wang, and X.-Y. Wang, "A medical data sharing model via blockchain," *Acta Automatica Sinica*, vol. 43, no. 9, pp. 1555–1562, 2017.

[9] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and trustable electronic medical records sharing using blockchain," in *Proceedings of the AMIA Annual Symposium Proceedings*, pp. 650–659, Washington, DC, USA, November 2017.

[10] D. Di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable Access Control systems," *Computers & Security*, vol. 84, no. 6, pp. 93–119, 2019.

[11] Y. Zhang, M. Cui, L. Zheng et al., "Research on electronic medical record access control based on blockchain," *International Journal of Distributed Sensor Networks*, vol. 15, no. 11, pp. 1–13, 2019.

[12] F. Tang, Y. Chen, and Z. Feng, "Electronic prescription sharing scheme based on blockchainand proxy Re-encryption," *Computer Science*, vol. 48, pp. 498–503, 2021.

[13] J. Ma, H. Xue, F. Wang et al., "A data access control method based on blockchain," *Journal of Physics: Conference Series*, vol. 1828, no. 1, pp. 012113–012121, 2021.

[14] X. Cheng, F. Chen, D. Xie, H. Sun, and C. Huang, "Design of a secure medical data sharing scheme based on blockchain," *Journal of Medical Systems*, vol. 44, no. 2, p. 52, 2020.

[15] F. Chen, Y. Luo, J. Zhang et al., "An infrastructure framework for privacy protection of community medical internet of things," *World Wide Web*, vol. 21, no. 1, pp. 33–57, 2018.