Computers, Materials & Continua
DOI:10.32604/cmc.2020.012205
Article

Tech Science Press

Secure and Efficient Data Storage and Sharing Scheme Based on Double Blockchain

Lejun Zhang[1,2,*], Minghui Peng[1], Weizheng Wang[3], Yansen Su[4], Shuna Cui[5,6] and Seokhoon Kim[7]

[1]College of Information Engineering, Yangzhou University, Yangzhou, 225127, China
[2]School Math & Computer Science, Quanzhou Normal University, Quanzhou, 362000, China
[3]Division of Computer Science, University of Aizu, Aizu–Wakamatsu, 9658580, Japan
[4]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei, 230601, China
[5]Medical College of Yangzhou University, Yangzhou, 225001, China
[6]Department of Gynecology and Obstetrics, Affiliated Hospital of Yangzhou University, Yangzhou, China
[7]Department of Computer Software Engineering, Soonchunhyang University, Asan, Korea
[*]Corresponding Author: Lejun Zhang. Email: zhanglejun@yzu.edu.cn
Received: 19 June 2020; Accepted: 19 July 2020

Abstract: In the digital era, electronic medical record (EMR) has been a major way for hospitals to store patients' medical data. The traditional centralized medical system and semi-trusted cloud storage are difficult to achieve dynamic balance between privacy protection and data sharing. The storage capacity of blockchain is limited and single blockchain schemes have poor scalability and low throughput. To address these issues, we propose a secure and efficient medical data storage and sharing scheme based on double blockchain. In our scheme, we encrypt the original EMR and store it in the cloud. The storage blockchain stores the index of the complete EMR, and the shared blockchain stores the index of the shared part of the EMR. Users with different attributes can make requests to different blockchains to share different parts according to their own permissions. Through experiments, it was found that cloud storage combined with blockchain not only solved the problem of limited storage capacity of blockchain, but also greatly reduced the risk of leakage of the original EMR. Content Extraction Signature (CES) combined with the double blockchain technology realized the separation of the privacy part and the shared part of the original EMR. The symmetric encryption technology combined with Ciphertext-Policy Attribute-Based Encryption (CP–ABE) not only ensures the safe storage of data in the cloud, but also achieves the consistency and convenience of data update, avoiding redundant backup of data. Safety analysis and performance analysis verified the feasibility and effectiveness of our scheme.

Keywords: Cloud storage; blockchain; electronic medical records; access control; data sharing; privacy

1 Introduction

With the rapid development of information technology, medical data has become the key to discovering and treating diseases [1]. More and more data is transferred from paper to electronic equipment because of the digitization of electronic storage [2,3]. EMR has been a major way for hospitals to store patients' medical data. The emergence of EMR has brought great opportunities to the development of wise medical practice [4]. Because the value inherent in EMR has given birth to business entities [5,6], EMR sharing is considered to be a promising approach [7,8]. However, there are few critical problems in this environment. 1) It is difficult for patients to obtain the data stored in the hospital [9]. 2) The conventional solutions are still vulnerable to information loss [10]. 3) Different medical institutions are loath to share their data [11]. To address these issues and meet the high demands on data sharing [12], some researchers proposed to use a third–party cloud instead of a private database for data sharing [13,14], and some cryptographic schemes have been proposed to solve these issues, though the disadvantages still exist [15,16]. For the storage and sharing of EMR, there are still some challenges, such as interoperability [17], data security, and privacy [18,19]. For the hospital, the sheer volume of data stored with third parties is not reassuring [20]. The consistency and interoperability of the different types of data from different medical institutions are big problems for data sharing [21]. The emergence of blockchain ensure security and transparency [22]. In recent years, the distributed healthcare blockchain system [23] has emerged [24,25].

Although the emergence of blockchain provides the possibility to solve these issues, the storage capacity of blockchain is limited and single blockchain schemes have poor scalability and low throughput. To address these issues, we propose an EMR storage and sharing scheme based on double blockchain. The main contributions of this paper are summarized as follows:

1. CES combined with the double blockchain technology realizes the separation of the privacy part and the shared part of the original EMR.
2. Cloud storage combined with the double blockchain technology not only solves the problem of the limited storage capacity of the blockchain and reduces the risk of medical data leakage, but also improves throughput and enhances scalability.
3. The symmetric encryption technology combined with the CP–ABE technology not only ensures the storage security of data in the cloud, but also achieves the consistency and convenience of data update.

The rest of the article is organized as follows: in Section 2, we review the related work about the storage and sharing of EMR, and then discuss their limitations. The related technologies of this paper will be described in Section 3. Next in Section 4, the system model of this paper will be described. In Section 5, the process of EMR storage, sharing and management in this scheme will be described in detail. In Section 6, we will conduct security analysis and performance analysis on our scheme. Finally, Section 7 concludes the paper and illustrates future expansion.

2 Related Work

In this section, we outline the research status of cloud services and blockchain technology to achieve secure storage and efficient sharing of EMR.

Zhang et al. [26] propose a secure medical record storage and sharing scheme based on double blockchain. In this article, patients encrypt their EMR with private keys and store them in a third-party cloud server. In fact, there is no reliable third party in the real world. The system designed by Xi et al. [27] is based on a permissioned blockchain which allows access to invited users and verified users. The strict access control reduces the efficiency of EMR sharing. The above two schemes both have the same problem in the sharing process of EMR. When a third party needs to view part of the EMR, the compete

EMR must be transmitted. It is easy to leak the privacy of the patient and cause waste of resources. To solve this problem and improve efficiency, scientific researchers have proposed CES. Liu et al. proposed a blockchain—based privacy—preserving data sharing scheme [28]. This scheme uses CES to achieve the separation of the private part and shared part of EMR. After patients remove the private part of the EMR, each shared part is encrypted and uploaded to the cloud, and the indexes are stored in the blockchain. Because the cloud only stores the shared part, the patient cannot retrieve the complete EMR.

The traditional centralized medical system and semi-trusted cloud storage are difficult to achieve a dynamic balance between privacy protection and data sharing. The storage capacity of blockchain is limited and single blockchain schemes have poor scalability and low throughput. To address these issues, we propose a secure and efficient medical data storage and sharing scheme based on double blockchain. In our scheme, we encrypt the original EMR and store it in the cloud. The storage blockchain stores the index of the complete EMR, and the shared blockchain stores the index of the shared part of the EMR. Users with different attributes can make requests to different blockchains to share different parts of the EMR according to their own permissions.

3 Preliminaries

3.1 Blockchain

Blockchain technology is the basic technology of Bitcoin [29] invented by the mysterious Satoshi Nakamoto in 2008. The block header contains information such as version number, previous block hash, nonce, Merkle root, timestamp and target difficulty. The blockchain operates in a peer-to-peer manner. After all transactions are broadcast in the blockchain network, all transactions will be allocated to each network maintenance node in the blockchain for verification. Only when 51% of the participating nodes in the blockchain network reach a consensus can the block be validated and added to the blockchain. All legal transactions are stored in data blocks. The basic structure of the blockchain is shown in Fig. 1.

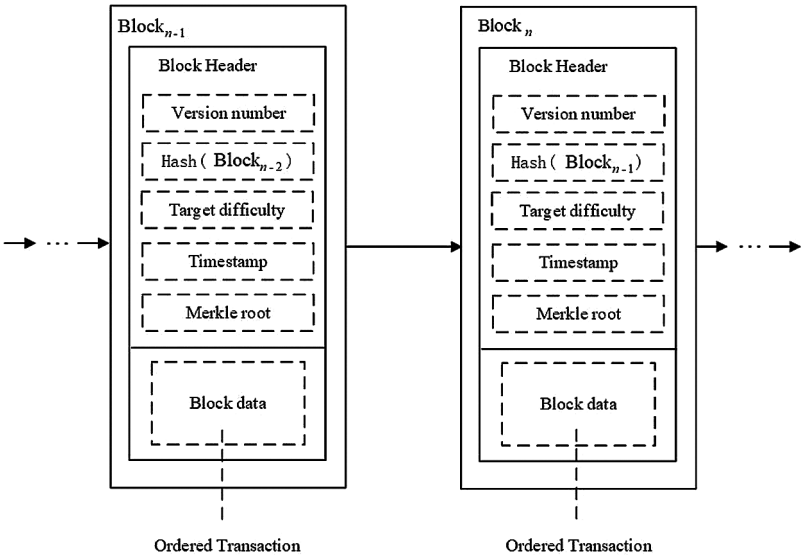

Figure 1: Blockchain basic structure

3.2 Smart Contract

The concept of smart contract was first proposed in 2014. Although the idea of smart contracts was proposed long ago, it has never been able to be implemented. It was not until the emergence of blockchain technology that it provided a supportable platform for smart contracts. Smart contracts are modular, reusable, and automatically executed scripts that run on the blockchain. Once the preset conditions are met, the smart contract can be performed automatically without a third party, and the results are written into the blockchain. Through using smart contracts, we can achieve trusted transactions, and these transactions are traceable and irreversible. For users who violate smart contracts, the smart contract setter has the right to revoke the user's authority.

3.3 Content Extraction Signature

When a third party needs to view part of the EMR, the compete EMR must be transmitted. But it is easy to leak patients' privacy and cause waste of resources. Therefore, there is a need for a digital signature scheme based on fine-grained level which must ensure that users can sign at any granular level and the signer can control the extraction method of the signed content. CES can meet the above requirements, and this method is more efficient in terms of computation and communication. CES allows users to remove private data according to their wishes and extract the shared data [30]. It has been widely used in many fields.

3.4 Ciphertext Policy Attribute Based Encryption

The concept of attribute-based encryption not only realizes one-to-many communication means, but also enhances the information confidentiality. The attribute encryption mechanism is divided into Key Policy Attribute Based Encryption (KP–ABE) and CP–ABE. The specific process is described as follows: Firstly, the authority sets public parameters and master key. Secondly, the data owner can define his own access control policy. The ciphertext adopts a tree structure to describe the access policy. Thirdly, the data owner encrypts the message to form a ciphertext. Fourthly, after users submit their attributes to the certification authority, they will obtain their own public key and private key. Finally, only when their attributes satisfied the access policy, the user can decrypt the ciphertext.

4 EMR Storage and Sharing Model Based on Double Blockchain

4.1 Notations

Notations and corresponding descriptions are given in Tab. 1.

Table 1: Notations

Notations	Description
P	Patient
D	Doctor
$U/U_p/U_g$	User/privileged user/general user
PK_{doc}/SK_{doc}	D's key pair for CES
PK_{pat}/SK_{pat}	P's key pair for CES
CN/MN	Consensus node/master node
PK_{CN}/SK_{CN}	CN's key pair
PK_{MN}/SK_{MN}	MN's key pair
K_{doc}	D's symmetric encryption key

Table 1 (continued).

Notations	Description
K_{pat_i}	P's symmetric encryption key
$M_{share}/M_{private}/M_{full}/ext$	Shared part/privacy part/complete EMR/extraction part
δ_i /δ_{full} //δ_{ext}	Signature of sub-message/full signature/extract signature
A_U	U's attribute set
$SK_{A_U}/SK_{A_{U_g}}/SK_{A_{U_p}}$	$U//U_g/U_p$'s attribute private key
A_{C-CP}	The access policy
$Index_{share}/Index_{full}$	M_{share}'s index/M_{full}'s index
C_{full}	Ciphertext stored in the cloud
url_{full}	C_{full}'s storage address
PK	System public parameters
MK	System master key
CEAS	Content extraction access structure
H	A hash function
T	A timestamp
Tag_{share}/Tag_{full}	$M_{share's}$ tag/$M_{full's}$ tag
$A_{C-CP_{share}}/A_{C-CP_{private}}$	$M_{share's}$ A_{C-CP} /$M_{private's}$ A_{C-CP}

4.2 System Model

As shown in Fig. 2, our model is divided into three layers. The role of these three layers is introduced as follows.

Data Acquirement Layer. In the data acquirement layer, D generates M_{full} and δ_{full}. P can extract the sub-messages from M_{full}. After P uploads corresponding information to the cloud, the cloud will return url_{full}. P can generate $Index_{share}$ and $Index_{full}$ according to url_{full}.

Data Storage Layer. The main function of this layer is to store $Index_{full}$, $Index_{share}$ and C_{full}. We use the storage blockchain to store $Index_{full}$ and use the shared blockchain to store $Index_{share}$. The cloud stores the ciphertext of sub-message, the corresponding symmetric key ciphertext, and the signature of the sub-message.

Data Sharing Layer. We achieve the data sharing of M_{full} and M_{share}. U_p can obtain M_{full} after making a request to the storage blockchain. U_g can send requests to the sharing blockchain to achieve sharing M_{share}.

5 EMR Storage and Sharing Scheme Based on Double Blockchain

5.1 EMR Storage Based on Double Blockchain

5.1.1 EMR Storage Based on Double Blockchain

It is assumed that EMR contains 9 parts: Name, gender, date of birth, *ID* number, symptoms, diagnosis result, prescription, medical examination report and medical history, $M_{full} = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\}$, $M_{\text{private}} = \{m_1, m_8, m_9\}$, $M_{share} = \{m_2, m_3, m_4, m_5, m_6, m_7\}$. Because the complete signature generation requires key pair, we first introduce key pair generation method.

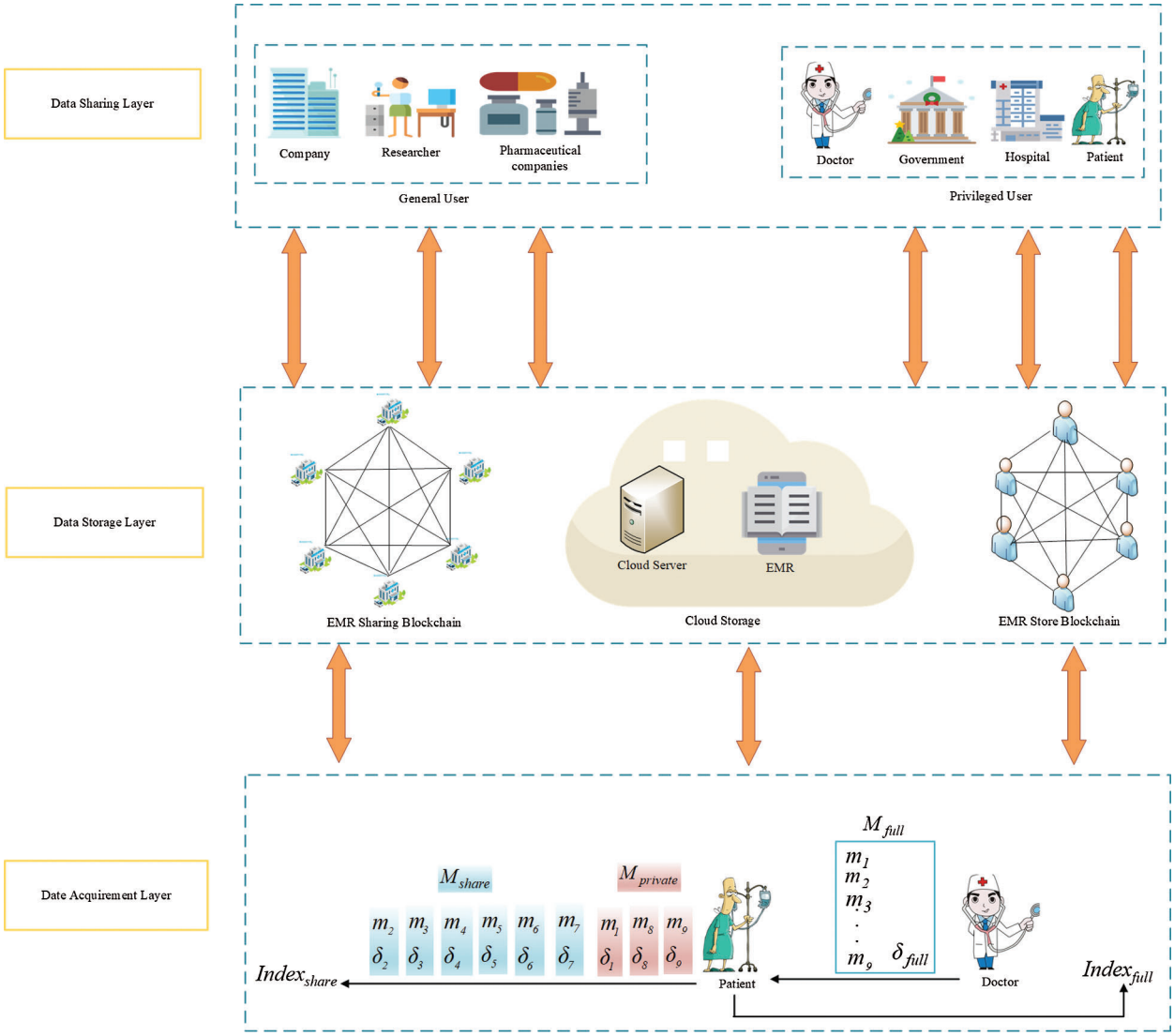

Figure 2: System overall framework

Key generation: Firstly, the certification authority randomly selects two unequal prime numbers: h and q. Secondly, the certification authority calculates $n = h \times q$ and sets Euler function: $\phi(n) = (h-1)*(q-1)$. Thirdly, D randomly selects an integer e that is prime with $\phi(n)$ in the interval $[1, \phi(n)]$, and find an integer d to satisfy $(e \times d) \bmod \phi(n) = 1$. Finally, according to the above calculation, $PK_{doc} = \{n, e\}$. $SK_{\text{doc}} = \{n, d\}$.

D will use the generated key pair to generate M_{full}'s δ_{full} based on the complete signature generation algorithm.

Algorithm name: The complete signature generation algorithm

Input: D's private key, $SK_{\text{doc}} = \{n, d\}$; P's EMR, M_{full}

Output: The full signature, δ_{full}

1) int $i = 1$; //Parameter for cyclic control

2) for (i = 1; i <= 9; i ++) {

3) Select a CES–Tag randomly with a fixed length, defined as r_i;}

4) for (i = 1; i <= 9; i ++) {

5) $H_i = H(m_i||r_i)$;} //The symbol || stands for connection, calculate the hash value of the sub-message connected with random number

6) $H = H_1||H_2||H_3||H_4||H_5||H_6||H_7||H_8||H_9$; // Connect the values of H_i together

7) $R = r_1||r_2||r_3||r_4||r_5||r_6||r_7||r_8||r_9$; //Connect the values of r_i together

8) $\delta_H = H\text{^}d \ mod \ n$; //Sign a with SK_{doc}

9) $\delta_{full} = \{\delta_H, R\}$;

10) return δ_{full};

In order to ensure the transmission security of data, D will use his symmetric encryption key K_{doc} to encrypt $(M_{full}\big|\big|H_{i\ (i \in [1,9]\ )}\big|\big|\delta_{full}||R)$, and use PK_{pat} to encrypt K_{doc}. Then D sends the set of two encrypted information $Info1$ to P, the set formula is shown in Eq. (1).

$$Info1 = \{E_{K_{doc}}(M_{full}\big|\big|H_{i\ (i \in [1,9]\ )}\big|\big|\delta_{full}||R), E_{PK_{pat}}(K_{doc})\} \quad (1)$$

5.1.2 Extraction of Sub-Messages

In this section, CES realizes the separation of the privacy part and the shared part. After P receive the set of two encrypted information $Info1$, P first decrypt ciphertext of K_{doc} with SK_{pat} and obtains $(M_{full}\big|\big|H_{i\ (i \in [1,9]\ )}\big|\big|\delta_{full}||R)$ further. In order to ensure the integrity and authenticity of the data, P will verify the correctness of δ_{full}. 1) For each sub-message m_i, P calculate the hash value $H(m_i||r_i)$, where $i \in [1,9]$. P determine whether the calculated hash value is equal to the hash value obtained in the decrypted message. If they are equal, go to step two. 2) P verify the correctness of $\delta_H = SIG(H; SK_{doc})$ using $PK_{doc} = \{n, e\}$ and calculate $\delta_H\text{^}e$ mod n, If the calculation result is equal to H, δ_H is a valid signature.

After P ensure that EMR and δ_{full} is accurate, P can extract each sub-message from EMR. We assume that the subset to be extracted is defined as X. The extraction signature generation algorithm is as follows. P can generate a signature corresponding to the sub-message according to the extraction signature generation algorithm. Therefore, the corresponding signature of the privacy part is $\{\delta_1, \delta_8, \delta_9\}$, and the corresponding signature of the shared part data is $\{\delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7\}$.

Algorithm name: The extraction signature generation algorithm

Input: P's EMR, M_{full}; EMR's full signature, δ_{full}; The subset to be extracted, X

Output: Extract signature of subset to be extracted, δ_{ext}

1) int i = 1; //Parameter for cyclic control

2) H_{unext} = null; //H_{unext} represents the hash value of the unextracted message, the initial value is null.

3) R_{ext} = null; //R_{ext} represents the random number of the extracted message, the initial value is null.

4) for (i = 1; i <= 9; i ++) {

5) Extract r_i from δ_{full};}

6) for (i = 1; i <= 9; i ++) {

7) if ($m_i \notin X$) {

8) $H_i = H(m_i||r_i)$;

9) $H_{unext} = H_{unext}||H_i$; }

10) else {

11) $R_{ext} = R_{ext}||r_i;\}\}$

12) $\delta_{ext} = \{\delta_{full}, H_{unext}, R_{ext}\}$;

13) return δ_{ext}

5.1.3 Encryption of Sub-Messages

In this section, the symmetric encryption technology combined with CP–ABE not only ensures the safe storage of data in the cloud, but also achieves the consistency and convenience of data update, avoiding redundant backup of data. After P extract each sub-message from EMR and generate an extraction signature corresponding to the sub-message, two different encryption methods need to be performed in sequence.

Firstly, P will use different symmetric encryption keys to encrypt each sub-message. The encryption method is shown in Eq. (2).

$$E_{K_i}(m_i) = C_{m_i}, \quad i \in [1,9] \tag{2}$$

Secondly, P set different access control policies for different symmetric encryption keys K_i, where $i \in [1,9]$, K_i's $A_{C-CPprivate}$ is shown in Fig. 3, where $i \in \{1,8,9\}$. K_i's $A_{C-CPshare}$ is shown in Fig. 4, where $i \in \{2,3,4,5,6,7\}$. The ciphertext C_{K_i} is generated as shown in Eq. (3), where i $\in \{1,8,9\}$. The ciphertext C_{K_i} is generated as shown in Eq. (4), where i $\in \{2,3,4,5,6,7\}$.

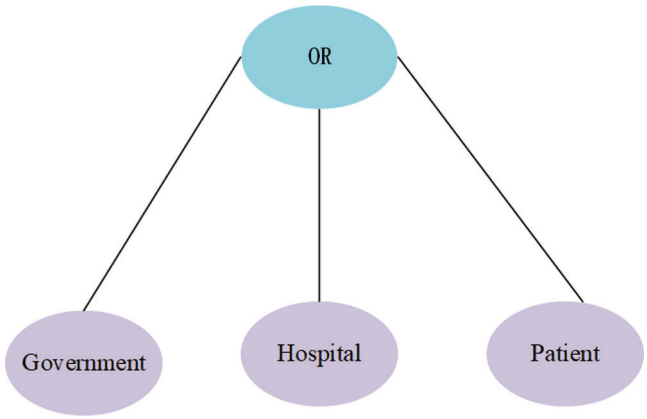

Figure 3: K_i's tree structure access control policy, $i \in \{1,8,9\}$

After P uploads the ciphertext to the cloud, P will receive the storage address url_{full}. The ciphertext stored in the cloud is shown in Eq. (5).

$$E_{PK}(K_i, A_{C-CPprivate}) = C_{K_i}, \quad i \in \{1,8,9\} \tag{3}$$

$$E_{PK}(K_i, A_{C-CPshare}) = C_{Ki}, \quad i \in \{2,3,4,5,6,7\} \tag{4}$$

$$C_{full} = \{C_{m_i}, C_{K_i}, \delta_i\}, \quad i \in [1,9] \tag{5}$$

5.1.4 Index Generation

In this section, P generates indexes of M_{share} and M_{full} respectively according to url_{full}. The two index generation methods are introduced as follows.

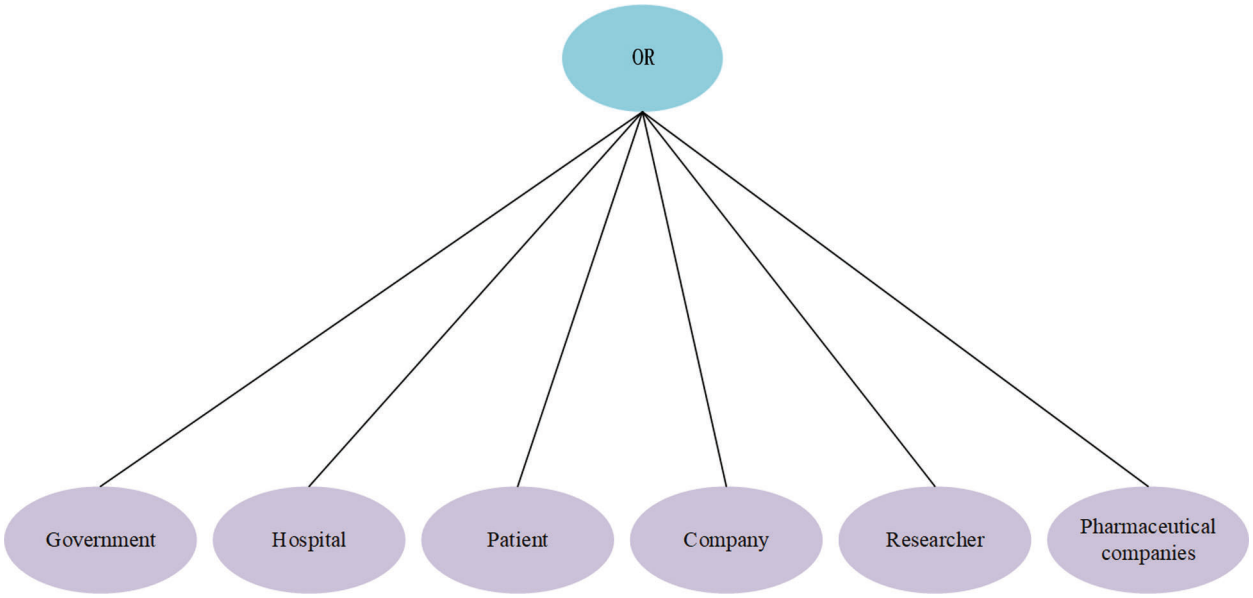

Figure 4: K_i's tree structure access control policy, $i \in \{2,3,4,5,6,7\}$

***Index*$_{share}$ generation:** After P receive url_{full}, P first use SK_{pat} to sign the two parts of url_{full} and Tag_{share}, the generated signature is defined as $\delta_{Index_{share}}$, then P use PK_{pat} to encrypt url_{full}, Tag_{share} and $\delta_{Index_{share}}$. The encryption result combined with ID generates $Index_{share}$. Finally, P store $Index_{share}$ to shared blockchain. $Index_{share}$ generation process is shown in Eq. (6).

***Index*$_{full}$ generation:** The process of $Index_{full}$ generation is similar to the process of $Index_{share}$ generation. $Index_{full}$ generation process is shown in Eq. (7).

$$Index_{share} = \{ID||E_{PK_{pat}}\ (url_{full}||Tag_{share}||\delta_{Index_{share}})\} \tag{6}$$

$$Index_{full} = \{ID||E_{PK_{pat}}\ (url_{full}||Tag_{full}||\delta_{Index_{full}})\} \tag{7}$$

5.1.5 Index Release

In this section, unlike traditional index release, A double blockchain structure is used to achieve index release. The storage blockchain stores the index of the complete EMR, and the shared blockchain stores the index of the shared part. The detailed process is described as follows. 1) P use SK_{pat} to sign $Index_{share}$ and $Index_{full}$ respectively and get the signatures $SIG_{SK_{pat}}\ (Index_{share})$ and $SIG_{SK_{doc}}\ (Index_{full})$. 2) After P generate the signature and calculating the index hash, P will send a request Req_{share} to the shared blockchain to request storage $Index_{share}$, the request generation process is shown in Eq. (8). P will also send a request Req_{full} to the storage blockchain to request storage $Index_{full}$, the request generation process is shown in Eq. (9).

$$Req_{share} = \{Index_{share}||H\ (Index_{share})\ ||SIG_{SK_{pat}}\ (Index_{share})\} \tag{8}$$

$$Req_{full} = \{Index_{full}||H\ (Index_{full})\ ||SIG_{SK_{doc}}\ (Index_{full})\} \tag{9}$$

In the next section, $Index_{share}$ and $Index_{full}$ are collectively referred to as $Index$. Req_{share} and Req_{full} are collectively referred to as Req. Consensus process is described as follows.

(1) The master node will verify and collect legal transactions in a data set $Data_{set}$, the generation method of the data set is shown in Eq. (10). The master node will broadcast $Recive$ to all consensus nodes for verification. The hash value of the data set is shown in Eq. (11), and the set $Recive$ is shown in Eq. (12).

$$Data_{set} = \{Req||T\} \tag{10}$$

$$Data_{hash} = H\ (Data_{set}||T) \tag{11}$$

$$Recive = \{Data_{set},\ \ Data_{hash},\ \ T,\ \ SIG_{SK_{MN}}\ (Data_{set}||Data_{hash})\} \tag{12}$$

(2) If more than 50% of the consensus nodes agree, this means that new blocks are successfully created, the data will be uploaded to the blockchain. D_{block} is shown in Eq. (13). The public key of the consensus node is defined as PK_{CN}.

$$D_{block} = \{Data_{set}||Data_{hash}||PK_{CN}||SIG_{SK_{CN}}(Res||T)||T\} \tag{13}$$

5.2 EMR Sharing Based on Double Blockchain

5.2.1 Double Blockchain Access Authentication

In order to achieve the sharing of EMR, first of all, patients first need to make a request to the blockchain to obtain the cloud data storage address. The steps to obtain the cloud data storage address are the same for both sharing M_{share} and sharing M_{full}. U initiates an EMR request transaction Req_U to the blockchain network. The request Req_U is shown in Eq. (14). Once the preset conditions are met, the smart contract can be performed automatically without a third party, and the results are written into the blockchain. Through using smart contracts, we can achieve trusted transactions, and these transactions are traceable and irreversible. For users who violate smart contracts, the smart contract setter has the right to revoke the user's authority.

$$Req_U = (ID\|T) \tag{14}$$

5.2.2 Ciphertext Acquisition

In this section, P will obtain the ciphertext stored in the cloud according to the obtained index. The detailed process is described as follows.

U_g first send a request to the sharing blockchain. If the request meets the access control preset by the smart contract, the smart contract will be automatically induced to use SK_{pat} to decrypt $Index_{share}$, then U_g can obtain url_{full}, Tag_{share} and $\delta_{Index_{share}}$. After U_g submits these data to the cloud server, the cloud server will verify the correctness of the signature, if the signature is correct, the cloud server will send the ciphertext C_{m_i}, the corresponding symmetric encryption key ciphertext C_{K_i} and the corresponding signature δ_i to U_g, where i $\in \{2, 3, 4, 5, 6, 7\}$.

The process of U_p requesting to share M_{full} is similar to U_g requesting to share M_{share}.

5.2.3 Ciphertext Decryption and Verification

In order to decrypt and verify the obtained ciphertext, U first submit A_U to the authorized institution. After the authorized institution verifies the accuracy of the attribute of U, the authorized institution will generate SK_{A_U}, then the authorized institution sends SK_{A_U} and PK to U. Therefore, the public key of U is PK and the private key is SK_{A_U}. Because different data users have different permissions, the decryption of ciphertext is divided into two parts: M_{share} ciphertext decryption and M_{full} ciphertext decryption. The following two detailed decryption processes are introduced as follows.

(1) M_{share} ciphertext decryption

After U_g obtain the shared part ciphertext C_{m_i}, the corresponding symmetric encryption key ciphertext C_{K_i} and the corresponding signature $\{\delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7\}$, where i $\in \{2, 3, 4, 5, 6, 7\}$. U_g will first decrypt the symmetric encryption key ciphertext, decryption process is shown in Eq. (15). Then U_g use the

obtained symmetric encryption key to decrypt M_{share} ciphertext, decryption process is shown in Eq. (16). Therefore, U_g have realized the sharing of M_{share}.

$$\text{Decrypt}_{SK_{A_{U_g}}}(C_{K_i}) = K_i, \quad i \in \{2,3,4,5,6,7\} \tag{15}$$

$$\text{Decrypt}_{K_i}(C_{m_i}) = m_i, \quad i \in \{2,3,4,5,6,7\} \tag{16}$$

(2) M_{full} ciphertext decryption.

After U_p obtain the shared part ciphertext C_{m_i}, the corresponding symmetric encryption key ciphertext C_{K_i} and the corresponding signature δ_i, where $i \in [1,9]$. U_p will first decrypt the symmetric encryption key ciphertext, decryption process is shown in Eq. (17), then U_p use the obtained symmetric encryption key to decrypt M_{full} ciphertext, decryption process is shown in Eq. (18). Therefore, U_p have realized the sharing of M_{full}.

$$\text{Decrypt}_{SK_{A_{U_p}}}(C_{K_i}) = K_i, \quad i \in [1,9] \tag{17}$$

$$\text{Decrypt}_{K_i}(C_{m_i}) = m_i, \quad i \in [1,9] \tag{18}$$

5.3 EMR Management Based on Double Blockchain

5.3.1 Definition of Sub-Message

In our scheme, P have the absolute right to use and control EMR. and patients should be able to redefine the privacy and shared parts according to their wishes.

Next a real scene will be described, assume that P need to define the medical examination report m_8 that was originally private data as shared data. In our scheme, P only need to change the original access policy from $A_{C-CPprivate}$ to $A_{C-CPshare}$, and then re-encrypt K_8 to obtain a new ciphertext $C_{NEW_{K_8}}$, encryption process is shown in Eq. (19). After obtaining the new ciphertext $C_{NEW_{K_8}}$, P only need to replace the original ciphertext C_{K_8} with the new one $C_{NEW_{K_8}}$. Through the above simple process, P can redefine the privacy and shared parts.

$$E_{PK}(K_8, A_{C-CPshare}) = C_{NEW_{K_8}} \tag{19}$$

5.3.2 Definition of User Rights

For U who violate the treaty, P should have the right to revoke U's authority. Therefore, the redefinition of user rights is very significant for P.

Next a real scene will be described, suppose P define the government that was originally U_p as U_g, and P define the researchers who were originally as U_g as U_p. In our scheme, K_i's access control policy need to change from $A_{C-CP_{private}}$ to $A_{C-CP_{NEW_{private}}}$, where $i \in \{1,8,9\}$. $A_{C-CP_{NEW_{private}}}$ is shown in Fig. 5. K_i's access control policy need to change from $A_{C-CP_{share}}$ to $A_{C-CP_{NEW_{share}}}$, where $i \in \{2,3,4,5,6,7\}$. $A_{C-CP_{NEW_{share}}}$ is shown in Fig. 6. P need to encrypt K_i to obtain a new ciphertext $C_{New_{K_i}}$ using a new access strategy $A_{C-CP_{NEW_{private}}}$, where $i \in \{1,8,9\}$. The encryption process is shown in Eq. (20), then P need to encrypt K_i to obtain a new ciphertext $C_{NEW_{K_i}}$ using a new access strategy $A_{C-CP_{NEW_{share}}}$, where $i \in \{2,3,4,5,6,7\}$. The encryption process is shown in Eq. (21). Through the above simple process, P can redefine user rights.

$$E_{PK}\left(K_i, A_{C-CP_{NEW_{private}}}\right) = C_{NEW_{K_i}}, \quad i \in \{1,8,9\} \tag{20}$$

$$E_{PK}\left(K_i, A_{C-CP_{NEW_{share}}}\right) = C_{NEW_{K_i}}, \quad i \in \{2,3,4,5,6,7\} \tag{21}$$

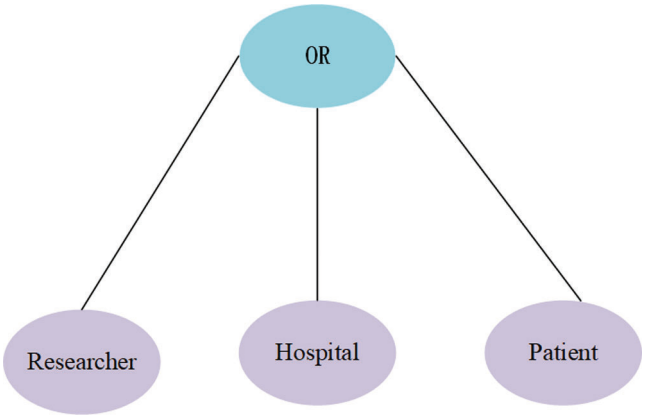

Figure 5: K_i's tree structure access control policy, $i \in \{1, 8, 9\}$

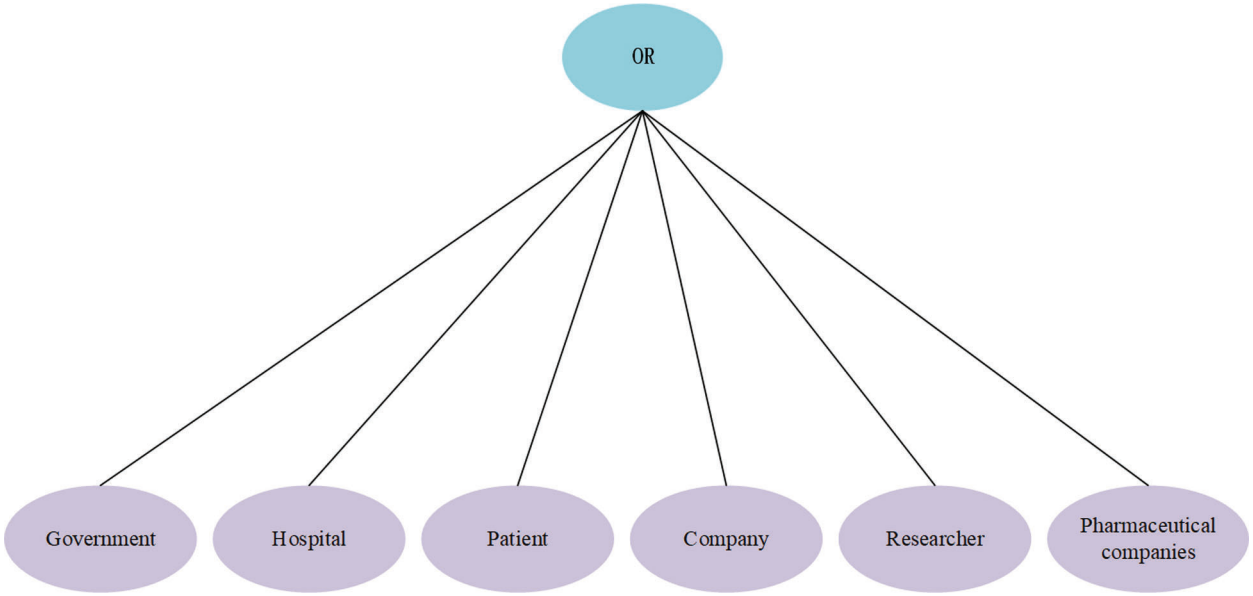

Figure 6: K_i's tree structure access control policy, $i \in \{2, 3, 4, 5, 6, 7\}$

6 Performance Analysis

6.1 Security Analysis

Security is a key issue in EMR sharing. Here, we analyze the security of our scheme from the following four aspects.

1. Anti-tampering: Our scheme encrypts the original EMR and stores it in the cloud, $Index_{full}$ is stored in the storage blockchain. $Index_{share}$ is stored in the shared blockchain. Therefore, the tamper-proof feature of the blockchain ensures that the original EMR stored in our cloud are immutable and cannot be modified arbitrarily.
2. Privacy protection: In our scheme, the semi-trusted cloud cannot obtain the plaintext of EMR. Compared with setting strict access control, P can separate the private part and shared part of the EMR according to their own wishes in our scheme, the double blockchain technology also realizes the separation of the private data and shared data. Our scheme achieves true privacy protection.

3. Data consistency: In order to realize that users with different attributes can access different parts, the cloud needs to store a complete EMR and a shared part in traditional schemes. While in our scheme, we use CP–ABE technology to encrypt M_{share} and $M_{private}$ separately so that the cloud only needs to store an original EMR, ensuring the consistency of data update and avoiding redundant backup of data.
4. Data integrity: From the generation of the complete EMR, to the extraction of the privacy and shared parts, and then to storage and sharing. Throughout these processes, our scheme ensures the integrity and accuracy of EMR.

6.2 Efficiency Analysis

6.2.1 Cloud Storage Efficiency Analysis

We compare the amount of data that the three schemes of traditional scheme, BPDS and our scheme need to store in the cloud, as shown in Tab. 2.

Table 2: Ciphertext storage

Scheme	Ciphertext storage
Traditional scheme	$C_{M_{full}},\ C_{M_{share}},\ \delta_{M_{full}},\ \delta_{M_{share}}$
BPDS	$C_{m_i},\ \delta_i,\ i \in [2,7]$
Ours	$C_{m_i},\ C_{K_i},\ \delta_i,\ i \in [1,9]$

From Tab. 2 we can see the comparison of the amount of data that the traditional scheme and our scheme. The cloud needs to store $C_{M_{full}}$, $C_{M_{share}}$, $\delta_{M_{full}}$ and $\delta_{M_{share}}$ in the traditional scheme. Our scheme only needs to store C_{m_i}, C_{K_i} and δ_i, where $i \in [1,9]$, there is no need to back up M_{share}. Although our scheme stores symmetric encryption ciphertext and signature of the key, the amount of these data is very small compared to M_{share}. Our scheme not only ensures the safe storage of data in the cloud, but also achieves the consistency and convenience of data update, avoiding redundant backup of data.

From Tab. 2 we can see the comparison of the amount of data between BPDS scheme and our scheme. BPDS scheme only needs to store C_{m_i}, δ_i in the cloud, where $i \in [2,7]$, while our scheme needs to stored C_{m_i}, C_{K_i}, δ_i in the cloud, where $i \in [1,9]$. Although our scheme stores more data, our scheme does not cause redundant backup of data. Compared with BPDS scheme, our scheme can realize that U_p can access M_{full} and U_g can access M_{share}.

It is assumed that the number of sub-messages in each complete EMR is z, the number of sub-messages in the shared part is x and the number of sub-messages in the privacy part is y. $z = x + y$. In the BPDS scheme, when P upload an EMR to the cloud, they need to upload the sub-messages of the shared part separately. But in our scheme, P only need to upload the EMR once.

The above results show that the amount of data that our scheme needs to store in the cloud is between the traditional scheme and BPDS scheme. Compared with the traditional scheme, our scheme saves storage space in cloud storage. Compared with BPDS scheme, our scheme saves the time of uploading EMR and the time of index generation, our scheme also provides better server quality.

6.2.2 Blockchain Storage Efficiency Analysis

When P need to store and share his own EMR, P need to store these x indexes into the blockchain in BPDS scheme. But in our scheme, P only need to store the index of the complete EMR in the storage blockchain, and store the index of the shared part in the shared blockchain. The storage capacity of the

blockchain is limited, our scheme better realizes the storage of the indexes and reduces the burden of blockchain storage.

6.2.3 Blockchain Sharing Efficiency Analysis

In BPDS scheme, all users request to share the shared part of the EMR on a blockchain. Because each sub-message of the shared part is separate, x request needs to be issued for the sharing of an EMR. But in our scheme, we adopt a tamper-proof double blockchain structure to distinguish users, at the same time, users only need to issue one request on a blockchain. Compared with BPDS scheme, our efficiency has improved a lot. The experimental results are shown in Figs. 7 and 8.

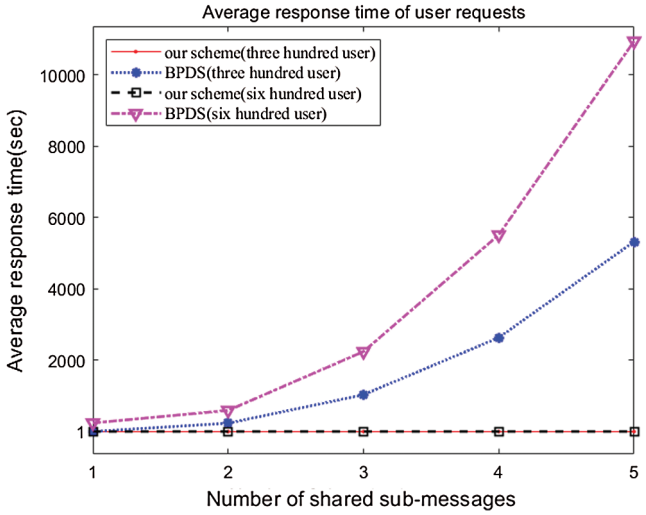

Figure 7: Response time of user requests when number of users and shared sub-messages is different

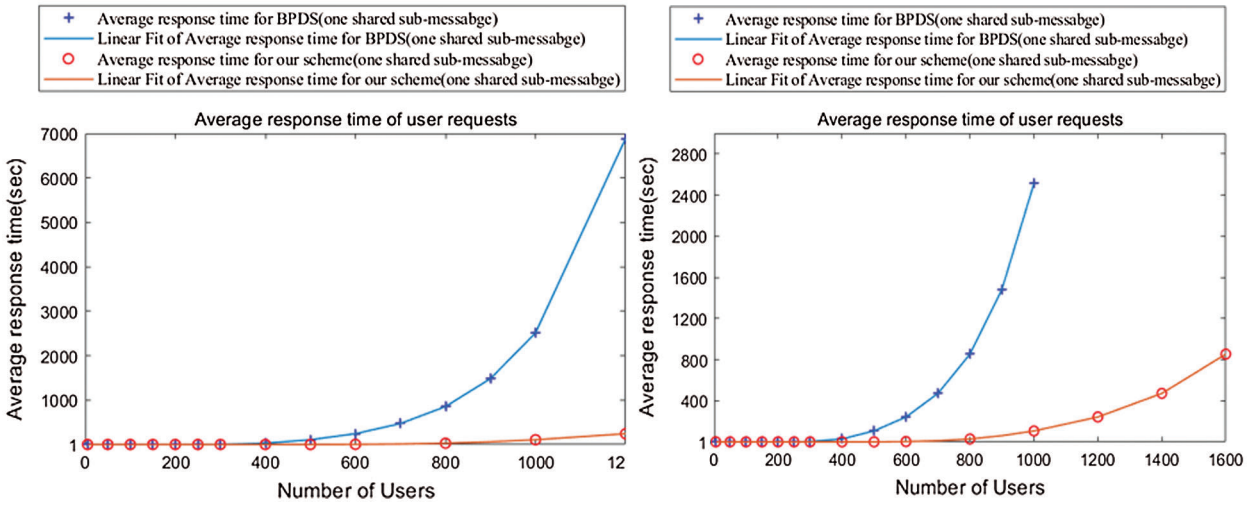

Figure 8: Average response time of user requests when number of users is different

As can be seen from Fig. 7, when the number of users in the system is fixed, as the number of sub-messages increases, the response time for processing user requests continues to increase in BPDS scheme. The reason is that each sub-message in BPDS scheme is separated. However, in our scheme, our

cloud stores a complete EMR, so the response time for processing user requests is not affected by the number of sharing sub-messages.

As can be seen from Fig. 8, when the number of users does not exceed the carrying capacity of the system, our scheme and BPDS scheme can still quickly process user requests, but as the number of users increases, the double blockchain structure has more and more obvious advantages compared with the traditional single blockchain, in the meanwhile, the double blockchain structure allows users with different attributes to request sharing on different blockchains.

6.2.4 Analysis of EMR Update Efficiency

(1) Definition of sub-message: When *P* need to redefine whether the sub-message of the EMR belongs to private data or shared data in BPDS scheme, *P* cannot reset the privacy and shared parts of EMR. But in our scheme, after extracting all the sub-messages of the EMR, *P* only need to re-encrypt the symmetric encryption key corresponding to the sub-message, and replace the original symmetric encryption key ciphertext.

(2) Definition of user rights: EMR that has been uploaded to the cloud should exist as a kind of historical data. When *P* need to redefine user rights, *P* need to re-encrypt the EMR and upload them to the cloud in BPDS scheme. The replacement of the original EMR will often lead to inconsistencies in the data. But in our scheme, re-encrypting symmetric encryption keys can realize the redefinition of user rights without changing and replacing the original EMR, our scheme ensures data update consistency and convenience.

7 Conclusion

The traditional centralized medical system and semi-trusted cloud storage are difficult to achieve a dynamic balance between privacy protection and data sharing. The traditional EMR storage and sharing scheme based on a single blockchain has poor scalability and low throughput. Our paper proposes an EMR storage and sharing scheme based on double blockchain. The original EMR is encrypted and stored in a semi-trusted cloud. We use a tamper-proof double blockchain structure to store the index of the complete EMR and the index of the shared part. Double blockchain structure allows users with different attributes to request sharing on different blockchains. CES combined with CP–ABE allows *P* to share EMR according to their wishes. Through experimental analysis, compared with the traditional scheme and BPDS scheme. our scheme not only saves the cloud storage space, but also ensures the consistency of storage. In our scheme, *P* can achieve complete privacy protection when sharing data.

Acknowledgement: The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper.

Funding Statement: This work is sponsored by the Natural Science Foundation of Heilongjiang Province of China under Grant No. LC2016024. Natural Science Foundation of the Jiangsu Higher Education Institutions Grant No. 17KJB520044 and Six Talent Peaks Project in Jiangsu Province No. XYDXX–108.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia *et al.,* "GridMonitoring: Secured sovereign blockchain based monitoring on smart grid," *Access IEEE*, vol. 6, pp. 9917–9925, 2018.

[2] G. Perera, A. Holbroo and L. Thdbane, "Views on health information sharing and privacy from primary care practices using electronic medical records," *International Journal of Medical Informatics*, vol. 80, no. 2, pp. 94–101, 2011.

[3] M. Turkanović, M. Hölbl, K. Košič, M. Heričko and A. Kamišalić, "EduCTX: A blockchain-based higher education credit platform," *Access IEEE*, vol. 6, pp. 5112–5127, 2018.

[4] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang *et al.,* "Consortium blockchain-based malware detection in mobile devices," *Access IEEE*, vol. 6, pp. 12118–12128, 2018.

[5] M. A. Uddin, A. Stranieri, I. Gondal and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *Access IEEE*, vol. 6, pp. 32700–32726, 2018.

[6] N. Tapas, G. Merlino and F. Longo, "Blockchain-based IoT-cloud authorization and delegation" in *SMARTCOMP*, pp. 411–416, 2018.

[7] C. Li, Y. Cao, Z. Hu and M. Yoshikawa, "Blockchain-based bidirectional updates on fine-grained medical data," in *2019 IEEE 35th Int. Conf. on Data Engineering Workshops*, Macao, IEEE, pp. 22–27, 2019.

[8] R. Guo, H. Shi, D. Zheng, C. Jing, C. Zhuang *et al.,* "Flexible and efficient blockchain-based ABE scheme with multi-authority for medical on demand in telemedicine system," *Access IEEE*, pp. 1, 2019.

[9] K. Fan, S. Wang, Y. Ren, H. Li and Y. Yang, "MedBlock: Efficient and secure medical data sharing via blockchain," *Journal of Medical Systems*, vol. 42, no. 8, pp. 136, 2018.

[10] J. Y. Zhang, S. Q. Zhong, T. Wang, H. C. Chao and J. Wang, "Blockchain-based systems and applications: A survey," *Journal of Internet Technology*, vol. 21, no. 1, pp. 1–14, 2020.

[11] Y. R. Ge, D. K. Ahn, B. Unde, H. D. Gage and J. J. Carr, "Patient-controlled sharing of medical imaging data across unaffliated healthcare organizations," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 157–163, 2013.

[12] S. J. Lee, E. B. Larson, S. Dublin, R. L. Walker, Z. Marcum *et al.,* "Electronic medical record (EMR) predictors of undiagnosed dementia," *Alzheimers & Dementia*, vol. 13, no. 7, pp. 1040–1041, 2017.

[13] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma *et al.,* "BlocHIE: A blockchain-based platform for healthcare information exchange," in *SMARTCOMP*, pp. 49–56, 2018.

[14] J. Huang, "A new economic model in cloud computing: Cloud service provider vs. network service provider," in *GLOBECOM*, pp. 1–6, 2015.

[15] R. Calvo, D. Thilakanathan and S. Chen, "Secure multiparty data sharing in the cloud using hardware-based TPM devices," in *2014 IEEE 7th Int. Conf. on Cloud Computing*, Anchorage, AK, IEEE, pp. 224–231, 2014.

[16] A. N. Khan, M. L. M. Kiah, M. Ali, S. A. Madani and A. U. R. Khan, "BSS: Block-based sharing scheme for secure data storage services in mobile cloud environment," *Journal of Supercomputing*, vol. 70, no. 2, pp. 946–976, 2014.

[17] N. Zeinali, A. Asosheh and S. Setareh, "The conceptual model to solve the problem of interoperability in health information systems," in *2016 8th Int. Sym. on Telecommunications*, Tehran, IEEE, pp. 684–689, 2016.

[18] A. Sajid and H. Abbas, "Data privacy in cloud-assisted healthcare systems: State of the art and future challenges," *Journal of Medical Systems*, vol. 40, no. 6, pp. 151, 2016.

[19] A. J. Burke, "Health information exchange (HIE) technology infrastructure for privacy assurance trustmark (PAT) test and development," in *Southeastcon*, 2015.

[20] B. Mishra and D. Jena, "Securing files in the cloud," in *2016 IEEE Int. Conf. on Cloud Computing in Emerging Markets*, Bangalore, IEEE, pp. 40–45, 2016.

[21] Z. Q. Xia, J. J. Tan, J. Wang, R. L. Zhu, H. G. Xiao *et al.,* "Research on fair trading mechanism of surplus power based on blockchain," *Journal of Universal Computer Science*, vol. 25, no. 10, pp. 1240–1260, 2019.

[22] W. J. Gordon and C. Catalini, "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability," *Computational & Structural Biotechnology Journal*, vol. 16, pp. 224–230, 2018.

[23] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf *et al.,* "On the security and performance of proof of work blockchains," in *Proc. of the ACM SIGSAC Conf. on Computer and Communications Security*, Vienna, Austria, pp. 3–16, 2016.

[24] A. Dubovitskaya, Z. G. Xu, S. Ryu, M. Schumacher and F. S. Wang, "Secure and trustable electronic medical records sharing using blockchain," in *Amia Annual Sym. proceedings*, Washington, D.C., IEEE, pp. 650–659, 2017.

[25] K. Peterson, R. Deeduvanu, P. Kanjamala and K. Boles, "A blockchain-based approach to health information exchange networks," in *Proc. of NIST Workshop Blockchain Healthcare*, Gaithersburg, MD, USA, pp. 1–10, 2016.

[26] N. Zeinali, A. Asosheh and S. Setareh, "The conceptual model to solve the problem of interoperability in health information systems," in *2016 8th Int. Sym. on Telecommunications*, Tehran, IEEE, pp. 684–689, 2016.

[27] X. Qi, S. Emmanuel and S. Abla, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information—An International Interdisciplinary Journal*, vol. 8, no. 2, pp. 44, 2017.

[28] J. Liu, X. Li, L. Ye, H. Zhang, X. Du *et al.,* "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," in *GLOBECOM*, Abu Dhabi, 2018.

[29] Y. Wang, A. Zhang, P. Zhang and H. Wang, "Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain," *Access IEEE*, vol. 7, pp. 136704–136719, 2019.

[30] S. J. Lee, E. B. Larson and S. Dublin, "Electronic medical record (EMR) predictors of undiagnosed dementia," *Alzheimers & Dementia*, vol. 13, no. 7, pp. 1040–1041, 2017.