doi: 10.20532/cit.2021.1005412

BPDST: Blockchain-Based Privacy-Preserving Data Sharing on Thin Client for Electronic Medical Records

Lu Xu[1], Mengchen Lin[1], Yong Feng[1] and Yani Sun[2]

[1]Yunnan Provincial Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming, China

[2]Yunnan Provincial Key Laboratory of Blockchain Application Technology, Yunnan Innovation Research Institute of Beijing University of Aeronautics and Astronautics, Kunming, China

Sharing medical data can improve the quality of medical services and reduce costs. However, the current Electronic Medical Records (EMRs) are scattered and easily tampered with, which is not conducive to the sharing of EMRs and is not compatible with thin clients. Fortunately, blockchain technology is tamper-proof, decentralized, auditable, and meets the above requirements. To solve these problems, we first propose Blockchain-Based Privacy-Preserving Data Sharing on Thin-Client for Electronic Medical Records (BPDST) approach that combines the k-anonymity and cloud storage, which thin clients can run like a full-node user and safeguard user's EMRs privacy concurrently. Using this approach, patients can control their own EMRs, while the consortium blockchain is responsible for the transaction process and sending the correct results to the patients. BPDST can also share information without leaking or tampering with EMRs' privacy, achieving the purpose of sharing medical data and privacy protection. In the medical field, this study can effectively protect users' privacy when sharing medical data to provide convenience for users and break the "island" phenomenon among various medical institutions. Security analysis and extensive experiments show that BPDST is secure and practical for sharing EMRs.

ACM CCS (2012) Classification: Security and privacy → Human and societal aspects of security and privacy

Keywords: consortium blockchain, EMRs, Privacy-Protecting, Thin-Client

1. Introduction

With the popularity of EMRs (Electronic Medical Records), the privacy problems in traditional EMR systems are becoming more and more obvious. For example, doctors may maliciously change patients' EMRs to avoid medical accidents. The reason for this problem is that EMRs are private data of patients, and doctors or hospitals can access or obtain these data at will. When doctors want to evade their responsibilities, EMRs will be maliciously changed, so that patients' privacy data cannot be guaranteed. As an important field of EMR development, smart medical management has received more and more attention in the management and sharing of EMRs. Meanwhile, many medical institutions, including hospitals, want to break the "island" phenomenon of EMRs. The so-called "island" phenomenon is that in the management of previous medical records, each medical institution managed its own EMRs, and there was no medium shared with the outside world. This greatly hinders the sharing of EMRs. Secondly, EMRs are difficult to store, although the relevant departments stipulate that medical data must be stored for a long time [1]. With the continuous increase of EMRs, undoubtedly, the management overhead of the hospital is a huge burden.

The "island" phenomenon may also lead to EMR information leakage or malicious tampering. With the development of cloud storage, many medical machines upload EMRs to a cloud-sharing center [2]. Cloud sharing can break the barriers of the "island" phenomenon, realize the sharing of EMRs among multiple medical institutions, and bring new opportunities for corresponding medical research and EMR management. Each medical institution can query and download EMRs in the cloud to conduct corresponding research under the condition of obtaining certification permission. However, cloud sharing has a prominent drawback. When the amount of data is very large, it is quite inconvenient to manage.

The emergence of blockchain has found a feasible breakthrough for EMR's shared privacy protection [3]. At present, there is research on information sharing and privacy protection based on blockchain at home and abroad. EMRs are a typical example of sharing data and protecting privacy, where blockchain can be used. Some studies are now focused on the distributed storage of EMRs in hospitals. Distributed storage can overcome the drawbacks of cloud storage and cloud-shared single-point attacks [4].

However, the current data sharing based on the privacy protection of the blockchain EMRs has an obvious flaw. The corresponding blockchain can only be run on full nodes. Because searching for data on the blockchain requires downloading a complete block, this is extremely unfriendly to mobile terminal devices [5]. The storage and computing performance of thin clients are quite limited. They cannot be mined like a full node, and they do not have the ability to download and store a complete blockchain. But now there is a large amount of data indicating that a considerable number of users operate the data information on the thin clients [6], which shows that the optimization for the thin clients is very necessary. The existing research on the privacy protection of EMRs is mainly divided into three directions: first, technical solutions, second, patients' concerns about the privacy of EMRs, and third, legal protection. The main users of EMRs are medical personnel. While EMRs are widely used, the protection of patient privacy must be strengthened simultaneously.

In this paper, we first present an approach called Blockchain-Based Privacy-Preserving Data Sharing on Thin Client for Electronic Medical Records (BPDST). The contribution of this paper is divided into the following three aspects:

- We creatively present the Blockchain-Based Privacy-Preserving Data Sharing on Thin Client for Electronic Medical Records (BPDST) approach in which blockchain is characterized by non-tampering, non-forgery, non-fiction, decentralization, *etc.* When attacked, it will not cause a single point of failure, the stored patient information will not be lost, and the uploaded patient-related data will not be maliciously tampered with. Readers are relevant medical institution personnel, patients, companies, government departments, and other institutions involved in obtaining patients' EMRs. And thin clients can run normally like a full-user node user.
- K-anonymity is applied to BPDST. When querying patient information, it hides the queried patients in k individuals to avoid exposing specific patients.
- In BPDST, the combination of on-chain and off-chain is adopted to relieve the storage pressure of the blockchain and break the "island" phenomenon.

The rest of the paper is organized as follows. We summarize the background knowledge and literature survey in section 2. In section 3, we introduce EMRs' data-sharing model based on blockchain. In section 4, we propose a data sharing based on EMR privacy protection of thin clients under the premise of blockchain. Then, the performance analysis is described in section 5. the discussion is described in section 6. Finally, we conclude the paper in section 7.

2. Background Knowledge and Literature Survey

2.1. Blockchain

Blockchain is a decentralized distributed ledger system that transfers, pays, and trades in a peer-to-peer manner [7]. The verified transactions are saved and connected in time sequence in the data block, as shown in Figure 1. Compared with the traditional centralized ledger system,

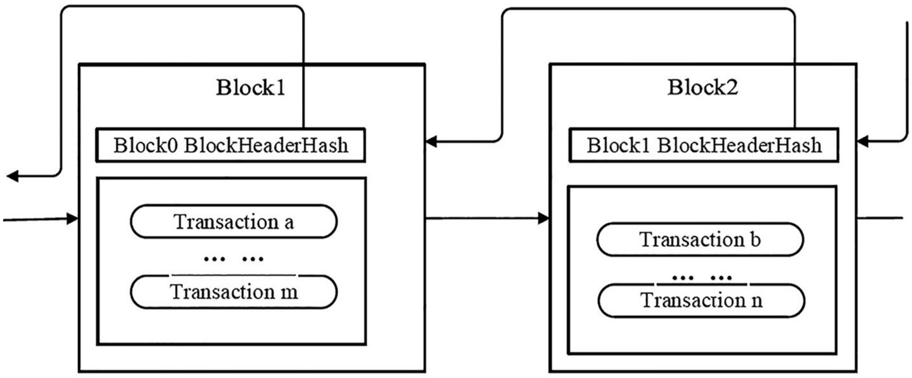

Figure 1. The Structure of Blockchain.

the blockchain has the advantages of complete disclosure, non-tampering, and prevention of multi-impact payment, and does not rely on any trusted third party [8]. There are three types of blockchains [9]:

1. Public chain: The public chain is open, and every Internet user can freely join it. After joining, they can read the data on the chain and send transactions and participate in the consensus process of the block.
2. Private chain: A private chain refers to a blockchain that is managed by an organization or institution and whose read permissions are not completely open to the public or are subject to certain restrictions. A private chain usually has a trusted center, which is a partially decentralized structure.
3. Consortium chains: Consortium chains are generally controlled by multiple centers. Relevant organizations shall cooperate to maintain a blockchain, which has restricted access with permissions. The consortium chain can be regarded as "partially decentralized", and the public can read and trade, but the validation of transactions or the release of smart contracts requires the consortium's permission.

2.2. Consensus Mechanism

The consensus mechanism is used to reach a consensus among nodes and link data. Its characteristics are "equality for all" and "subject to the majority". This chapter introduces three typical consensus mechanisms [10-12]:

1. Proof of work (POW), which takes the computational cost required to solve computationally difficult problems as the credential of the newly added block and obtains incentive income.
2. Proof of stake (POS) is more efficient than POW, which replaces the workload with proof of stake, and the node with the highest stake realizes the addition of new blocks and obtains incentive income.
3. Delegated proof of stake (DPOS). The difference between POS and DPOS is that POS is the stakeholder and DPOS is the representative selected by the stakeholder to verify and generate the block. Dishonest representatives are easily eliminated.

2.3. Medical Blockchain

Using blockchain technology to store EMRs, the patient is the full controller of the medical data, rather than the data being controlled by a hospital or other institution. For ordinary residents, digital files of health information can be made from birth, and no matter where they move in the future, they can use the blockchain to access the hospital. Blockchain brings a new change to the management of medical-related information and the dissemination of an organization's information. Patients can control their own health information independently, which is more beneficial to the privacy protection of patients and also enhances the willingness of patients to use medical data so that the patients' medical information can be freely shared, and medical services can be decentralized. Due to the development of networks and cloud computing, the combination of cloud platforms can expand the scope of sharing medical information on electronic health platforms, saving residents' time and effort in back-and-forth verification between different departments.

In the traceability management of medical products and medical devices, all medicines and devices can be supervised from the manufacturer to the consumer terminal. The acquisition of these data is entirely based on the decentralized high-efficiency and low-cost blockchain platform [13].

2.4. The K-Anonymity Algorithm

K-anonymity was proposed by Samarati and Sweeney in 1998. K-anonymity protects data by reducing the accuracy of published data. Each record will have the same identifier attribute value as the $(k - 1)$ record, thus reducing the risk of data privacy disclosure caused by link attacks [16].

K-anonymity has the following three characteristics:

1. the attacker cannot find the specific data to be attacked in the attack information;
2. in a piece of data, the attacker is not sure about the light and dark attributes of the attack data;
3. the attacker is unable to determine the specific owner of a piece of information.

2.5. Literature Survey

L. Hai *et al.* [14] proposed K-anonymity location privacy protection based on blockchain, which is not combined with EMRs. J. Fu *et al.* [15] proposed the privacy protection of the medical blockchain system, which did not consider thin clients. D. Wang [16] proposed privacy protection in cloud computing in which deleting identifiers to protect published data is not really a way to prevent privacy disclosure. Attackers can obtain personal privacy data through link attacks. T. Xue *et al.* [17] proposed a medical data-sharing model based on blockchain technology, but there are problems with the implementation of this design. J. Liu *et al.* [18] proposed privacy protection data sharing of EMRs on the blockchain did not consider the convenience of patients. S. Alexaki *et al.* [19] proposed the concept of using blockchain and smart contracts to manage electronic health records and sharing. The MedRec proposed by A. Azaria *et al.* [20] is an implementation of a case management system based on the Ethereum blockchain platform, but the system uses POW to reach consensus, which is expensive in terms of time cost. K .N. Griggs *et al.* [21] proposed EMRs privacy protection based on blockchain that does not consider the storage burden of blockchain. O. Gutiérrez *et al.* [22] proposed an IT architecture based on blockchain electronic medical records, without considering the implementation. S. Jayakumar *et al.* [23] proposed that the blockchain-based electronic health record system is not specific enough for privacy protection. J. Marquis *et al.* [24] proposed to put the blockchain into the electronic medical record but did not implement the process. Hang *et al.* [25] proposed a blockchain-based EMR integrity platform that does not consider privacy protection when EMRs are shared. S. G. Alonso *et al.* [26] proposed a new blockchain challenge in the field of electronic health, and no solution was applied to EMRs. R. Johari *et al.* [27] proposed blockchain technology for medical record security, which is just an overview and has not been designed in detail.

3. EMR Data Sharing Model Based on Blockchain

EMRs refer to the text, symbol, chart, graph, data, image, and other digital information generated by a medical institution's information systems during medical activities. The information is very sensitive and extremely private, which may include the patient's name, ID number, home address, and other private information. In the preservation of traditional medical records, medical institutions have absolute possession and management rights over them. When cloud storage has developed to a certain point, some medical institutions use cloud storage to digitize medical records. The current time limit for medical records is about 30 years [1]. According to China's current medical scale, if all EMRs are stored in the cloud, even distributed clouds, this will bring considerable storage pressure and transmission loss. Moreover, the central storage represented by cloud storage is easily affected by a single point of failure, which will bring considerable losses. So now there is a considerable part of research on EMR shared privacy protection based on blockchain [17-18, 28].

In the blockchain-based EMRs privacy protection scheme, the current typical model is to divide the system into a three-layer structure: EMR upload layer, EMR storage layer, and EMR shared layer, as shown in Figure 2.

The process of hyperledger fabric transaction processing is shown in Figure 3. The process is, as follows:

1. The application client uses fabric node SDK to send transactions to the consortium chain.
2. After receiving the transaction proposal, the endorsement node verifies its signature and sends the result to the application client.
3. After the application client receives the information returned by the endorsement node, the application client determines whether the proposal is consistent through the endorsement policy when deploying the smart contract, and then sends the result to the orderers.
4. The orderers sort the received transactions by consensus and then package a batch of transactions together according to the block generation strategy to generate new blocks and send them to the submit node.
5. After receiving the block, the submitting node will verify each transaction in the block and add the block to the local blockchain after completion.

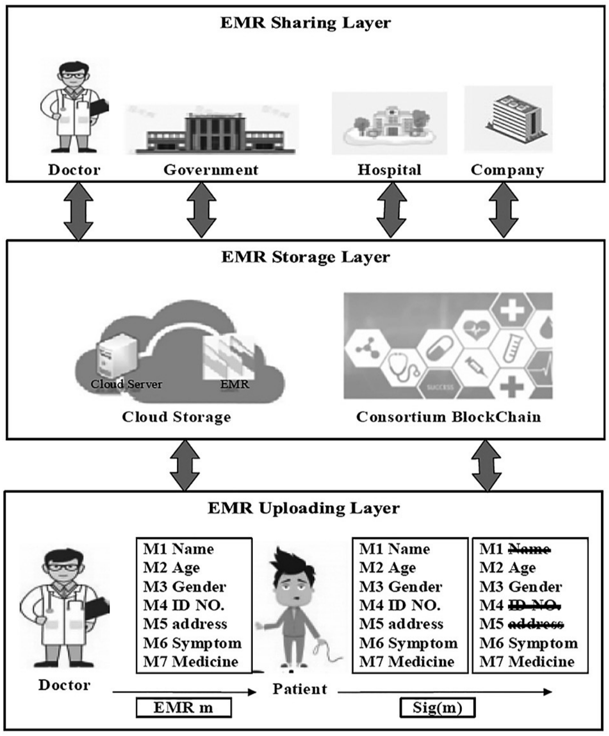

Figure 2. Three-layer Model of EMR Shared System.

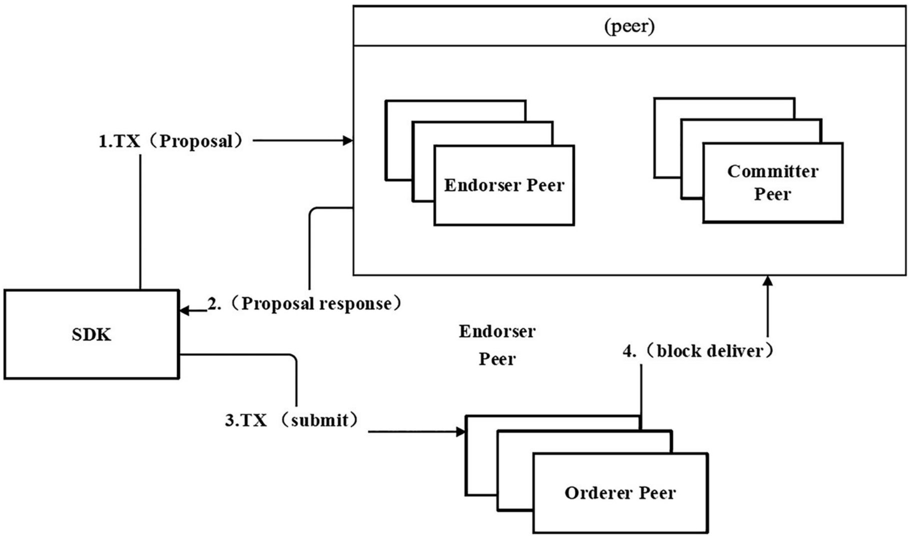

Figure 3. The process of transaction.

4. Blockchain-Based Privacy-Preserving Data Sharing on Thin Client for Electronic Medical Records Approach

4.1. System Design

Figure 4 shows the timing diagram of the EMR system. Currently, data sharing based on blockchain EMRs privacy protection is divided into three layers: EMRs upload layer, EMRs storage layer, and EMRs sharing layer. In this model, the flow direction and sequence of electronic medical records are marked by serial numbers in Figure 4. We assume that all patients operate by smartphones, that is, all patients are thin clients, mining operations cannot be performed independently, and consortium blockchain is required. It can be seen in Figure 2 that both the EMRs upload layer and the EMRs shared layer are associated with the EMRs storage layer. The patient uploads the processed electronic medical information to the cloud storage of storage layer, and the consortium blockchain is responsible for managing the electronic physiotherapy data uploaded to the cloud. Then, if an institution wants to query the relevant medical records, the patient's consent is required. Because the thin client has limited computing power, the patient seeks help from the consortium blockchain, finds the corresponding medical records, and sends them to the patient. If the patient and the institution use two-way authentication, the patient will release the medical records found to this institution. At this time, a complete sharing process is completed, that is, one upload and one query process.

The consortium chains have the advantages of high write throughput, data sharing with privacy protection, and support for consensus protocol expansion. Because EMRs have a wide range of sources, strong dynamics, and features of distributed management, we use a blockchain-based access control mechanism to manage user permissions, and the requester's access control information serves as a unique identity.

4.1.1. EMR Uploading Layer

In this layer, the original electronic medical data M is generated by doctor D and sent to patient P. As the owner and manager of M, the patient has the right to delete the sensitive information in M and upload it to the storage layer. We assume

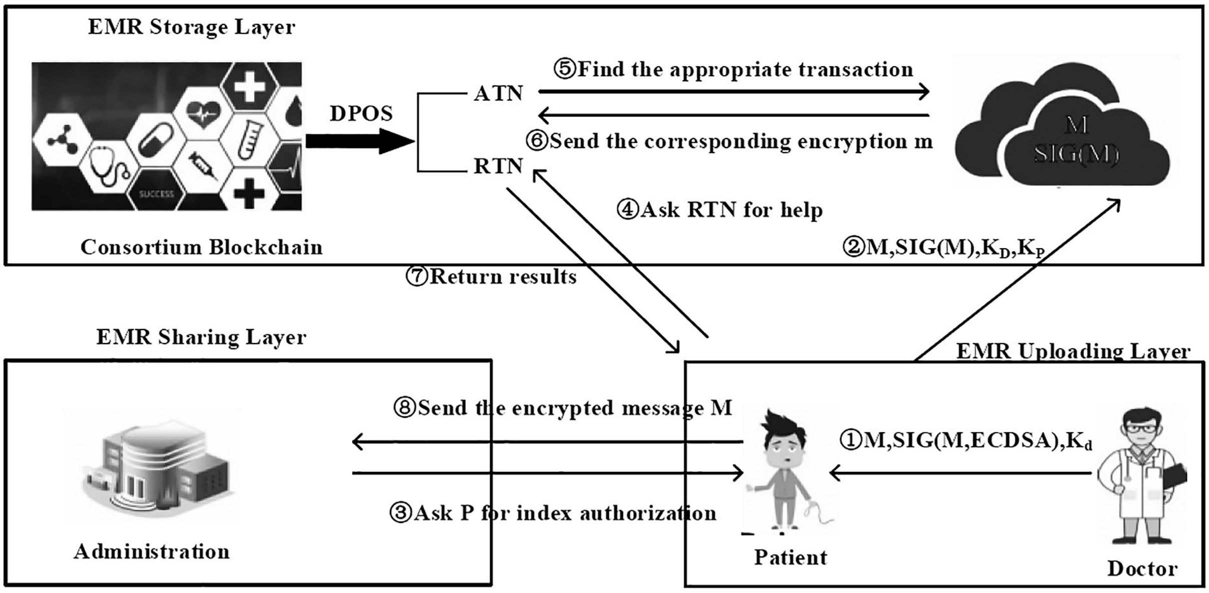

Figure 4. EMRs Shared System Timing Diagram.

that when a new patient P joins the system, all doctors and patient users previously assigned smart cards, and the blockchain assigned smart cards to the current new user P.

Doctor D transmits the corresponding original electronic medical data M to the current patient P. Based on the identity signature and encryption algorithm, the private key in the smart card is used to sign and the public key of the patient is used to encrypt the patient's information to generate the encrypted signature SIG (M, ECDSA), send the encrypted message to patient P, and then P uses the private key in the smart card to decrypt it to obtain the original EMRs information, as shown in Figure 4.

This article sets the patient to upload their own EMR records to the storage layer. The first reason is that the patient is the owner and administrator of EMRs, who have the right to choose whether to upload and share their own EMR records; the second reason is that when doctors generate EMR records, their medical institutions already have relevant EMR records. For the medical institution, no other upload operation is required.

After patient P obtains his original medical data M, he deletes the corresponding privacy part to generate processed medical data M' according to his own situation and then uses the signature and encryption algorithm to encrypt M's signature and upload it to the EMRs storage layer. Figure 4 shows the Signing Process in the EMR Uploading Layer.

4.1.2. EMR Storage Layers

In order not to leak data in EMRs, patients selectively delete private information before data is uploaded to the EMR storage layer. The cloud in the storage layer stores the encrypted EMR information M' and the corresponding signature SIG(M'). In the process of querying EMR information, the query path will be output and stored in the cloud at the same time, so that the tracker path can be tracked if it is maliciously modified. The EMRs in cloud storage will only be indicated by the index in the consortium blockchain, and users cannot query the EMRs' information stored in the cloud from the outside world or by other means.

We chose to use the consortium blockchain to save the index of EMR data and achieve data sharing. We set 101 medical institutions elected by DPOS as representative mining pools [29] and selected 30 institutions as representative nodes (RTN) and 20 institutions as audit nodes (ATN). The 30 RTNs are responsible for mining to help thin clients verify the legitimacy of querying user identities and find EMR index information. The responsibility of 20 ATNs is to supervise whether there are dishonest nodes in RTN and replace them if there are any.

The function of the storage layer is to store the uploaded EMR data and its corresponding index. The index contains the address and other related information stored by the EMRs. Currently, the signature of electronic medical data serves as an index for EMRs. This layer contains cloud storage and consortium blockchain. Cloud storage stores encrypted EMR information, and consortium blockchain stores corresponding index information, which contains encrypted EMR addresses, and other information stored in the cloud. As shown in Figure 5, the block in the consortium blockchain is composed of a block header and a block body. The index corresponding to the EMRs is stored in the block body, which contains information such as the storage address of the EMRs. Only when the corresponding block is found correctly, the correct index can be found, that is, the correct EMR data can be found. To find the corresponding EMR information, one can retrieve the corresponding EMR data in the cloud just by looking up the index stored in the blockchain.

According to the characteristics of the blockchain, the data stored in the blockchain can only be added and queried, and the added information cannot be modified or deleted. However, the EMR information stored in the cloud can be added, deleted, modified, and checked. However, no matter which operation is performed, the corresponding index must be found through the consortium blockchain to obtain the relevant EMR path, and the EMR operation in the cloud is completed.

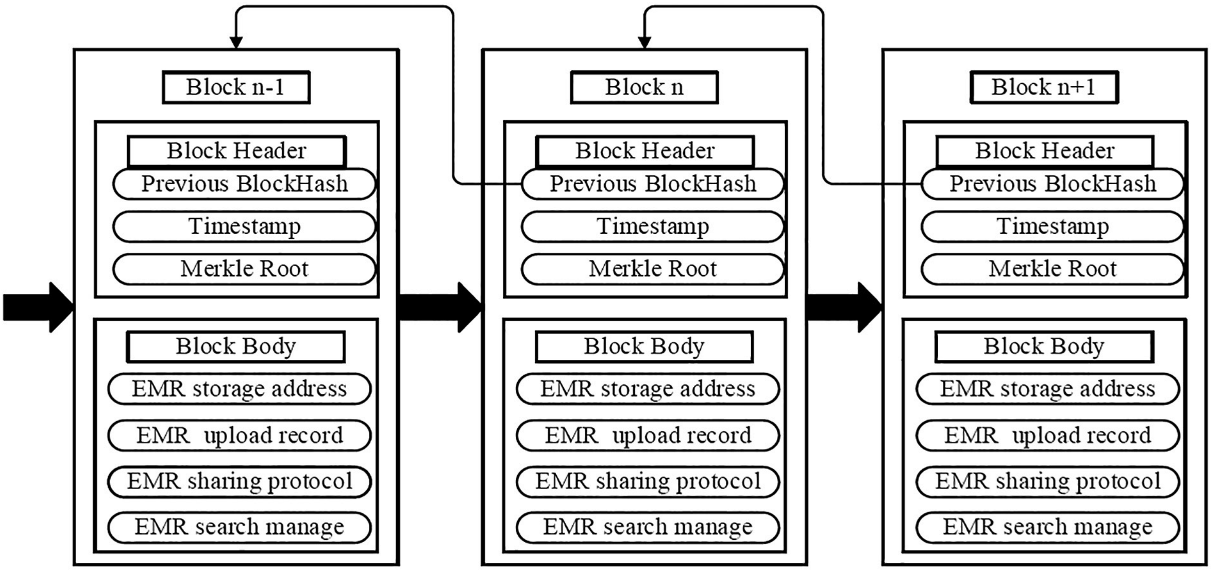

Figure 5. Storage Information in a Block.

4.1.3. EMR Sharing Layer

As shown in Figure 2, due to their respective needs, the legal identity of doctors, governments, and related institutions can query the EMRs in the system. Table 1 shows the relevant characteristics and functions defined by the basic parameters of the consensus mechanism we use in the consortium blockchain to ensure the legitimacy of the requester's identity, the authenticity of the requester, and the query information obtained.

Table 1. Corresponding parameters and functions of consortium blockchain.

Basic parameters	Function
Arbitration structure	Node information exchange in a pre-defined way.
Authentication	Verify the identity of the interrogator.
Completeness	Verify the integrity of the transaction.
Non-repudiation	The sender of the message must not deny sending the message.
Privacy	Ensure that only the inquirer can read the EMR information.
High fault tolerance	Failure of certain nodes does not affect the operation of the entire system.

Taking an insurance company as an example, if patient P applies for insurance company claims due to a certain disease, the insurance company needs to query P's EMR data in order to verify its authenticity. At this time, P is a thin client, and a complete node is needed to help it complete the mining process and to find the corresponding EMR. The patient needs to verify whether the insurance company's identity is legal, so as not to impersonate the attacker; then the insurance company also needs to verify the identity of the patient P to ensure that the received EMRs information is true and valid.

We can see from the above authentication process that at the shared layer, there is a two-way authentication process between the EMRs inquirer and the patient. While patient P verifies the legality of institution A's identity, institution A also needs to verify the authenticity and validity of patient P to ensure that the EMR data obtained is true and effective.

In traditional insurance claims, users first submit personal medical records, and then insurance companies go to their medical institutions for inquiries. Patients may provide false medical records, and insurance companies need to verify the authenticity of the medical records a second time. Therefore, traditional insurance claims have the disadvantages of relying on personal consciousness and complicated steps. In the BPDST model, you do not need to rely on personal consciousness, you can obtain the real EMRs through authentication; you do not need to verify the authenticity of EMR data twice, simplifying the steps in the claims process.

4.2. System Implementation

In our system, we adopt consortium blockchain fabric 1.4 to build our BPDST. Next, we will present the system implementation in detail. The notations and specific descriptions we need to use are shown in Table 2.

Table 2. Notations on BPDST used.

Notations	Descriptions
D_i	The *i*-th doctor
P_i	The *i*-th patient
CB	The consortium blockchain
ID_{Di}	The identity in blockchain
Sig(.) / *Ver*(.)	The ECC signature/verification
Enc() / Dec()	The AES-CBC encryption/ decryption
$Sk_{AES\text{-}CBC}$	The key of AES-CBC
Cer	The certification of signature
Q_i / d_i	The public/private key of ECC

4.2.1. Encryption and Signing Process

In order to implement the encryption and signature algorithm between the doctor and the patient in the EMRs upload layer, an additional RSA an AES-CBC encryption scheme is required. This public key encryption scheme has three characteristics:

1. The security of RSA algorithm is relatively high. It has a long key, and the amount of encryption computation is large.
2. RSA is an asymmetric algorithm. The encryption key is different from the decryption key. One key cannot deduce another key.
3. The security of AES-CBC is higher than that of ECB, which is more suitable for transmitting long messages and is not easy to be attacked.

Currently, we only have specific implementation proposals for signature schemes. In BPDST, we also propose a signature scheme that can avoid the privacy leakage caused by the transmission of public keys. The signature scheme used by BPDST is elliptic curve digital signature algorithm (ECDSA) that combines ECC and DSA. It is about twice the size of the public key bits required by the general elliptic curve encryption. The process of its signature and verification is: Suppose A wants to send a message to B. First, A and B need to select the elliptic curve and the origin G on the curve. Then, A will generate a random number d_A in the interval $[1, n-1]$, which is A's private key, and a public key $Q_A = d_A G$. If A wants to send message m, A will select a random number k in the interval $[1, n-1]$ and calculate:

$$\begin{gathered} Z = h(m),\ (x_1, y_1) = kG \\ r = x_1 \bmod n,\ s = k^{-1}(z + rd_A) \bmod n \end{gathered} \tag{1}$$

and then send the message m and the ECDSA signature(r, s) to B, and B receives it. The correctness of the ECDSA signature will be verified. First, B will calculate:

$$\begin{gathered} z' = h(m),\ u_1 = z's^{-1} \bmod n \\ u_2 = rs^{-1} \bmod n \\ (x_1', y_1') = u_1 G + u_2 Q_A,\ r = x_1' \bmod n \end{gathered} \tag{2}$$

If the verification passes, B confirms that the signature sent by A is correct. The process of signature verification is shown in Algorithm 1.

Algorithm 1. Signature verification contract.

```
Input: msg1, msg2
Output: Bool

If msg1, msg2 is not null then
    hash1 = sha256.Sum256( [] byte(msg1))
    hash2 = sha256.Sum256( [] byte(msg2))
    privBytes = ioutil.ReadFile(dir)
    key = Parse(privBytes)
    ecdsakey = key.(*ecdsa.PrivateKey)
    r, s = ecdsa.Sign(ecdsakey, hash1)
    cert = ioutil.ReadFile(dir)
    pubkey = cer.Publickey
If ecdsa.Verify(pubkey, hash1, r, s) is true &&
ecdsa.Verify(pubkey, hash2, r, s) is false then
    return true
End if
```

4.2.2. Registering and Uploading

To enter the BPDST system, one medical institution or doctor needs to first register in the consortium blockchain. Specifically, doctor Di with the identity of ID_Di will perform the following operations:

$$D_i \rightarrow CB: ID_{Di}, PK_{Di}, Q_i \qquad (3)$$

Upon receiving the $ID_{Di}||PK_{Di}||Q_i$, CB uses registered smart contract to check validity, displayed in Algorithm 2. After verification, CB computes a certificate $Cer_{CB,\,Di} = Sig_{CB}(T, ID_{Di}, PK_{Di}, A_i)$, where T represents the validity period of the certificate, and then CB replies to D_i through transaction:

$$CB \rightarrow D_i: T, Cer_{CB,\,Di} \qquad (4)$$

Finally, each patient P_i that uses the identity ID_{Pi} also registers at CB to obtain the certificate $Cer_{CB,\,Pi} = SigCB(T, ID_{Pi}, PK_{Pi})$.

Algorithm 2. Register contract of BPDST.

```
Input: $ID_{Di}$
Output: Bool

If Sender of message is not CB then
    Throw error to CB
End if
If $ID_{Di}$ has a problem in BPDST then
    Return the result false
If $ID_{Di}$ not exist of BPDST then
    $AuthorizeUsers[ID_{Di}] \leftarrow true$
    Return the result true
End if
```

During the treatment time, D_{i} generates the EMR_{Pi} within a treatment period for P_i. There EMR_{Pi} is encrypted as $\text{CEMR}_{Pi} = Enc_{Pi}(\text{EMR}_{Pi})$ by the EMRs sharing key K_{Pi} of P_i. Then P_i stores CEMR_{Pi} in the cloud and gets the hash value and index. Then, patient uploads the EMRs by uploading the EMRs information contract, as shown in Algorithm 3.

The order of contract deployment is:

peer chaincode install – *n* name – *v* 1.1 – *p* github.com/name, where name is a contract name deployed by CB.

Algorithm 3. Uploading the EMR information contract.

```
Input: EMRIndex, EMRHash, PatientID
Output: Bool

If PatientID is non-existent in BPDST then
    Throw error to CB
End if
If PatientID has a problem in BPDST then
    Return the result false
If PatientID has exist in BPDST then
    $uploadEMRInformation[PatientID] \leftarrow true$
    Return the result true
End if
```

4.2.3. Choice of RTN and ATN

In the EMR storage layer, we want to vote from the coalition pool of the consortium blockchain. The top 30 are used as RTNs, acting as complete nodes in the shared layer for mining operations; the next 20 nodes are used as ATNs to audit whether there are dishonest nodes in the RTN.

We use the DPOS consensus mechanism in the consortium blockchain. The block is created by a trusted account elected by the community. The trusted account is the trustee with the top 101 votes. Considering DPOS mechanism as a joint-stock company, ordinary shareholders cannot enter the board of directors or vote on their behalf [30]. If these representative mining pools use their rights to maliciously modify the data, the voters will immediately be kicked out of the BPDST system, and the substitute representatives can be replaced at any time.

Here we use a voting agreement with two central agencies in secure electronic elections [2]. One of the agreements is the Central Legal Authority (CLA) to prove the voters, and the other is a separate Central Tabulating Facility (CTF) to count the votes. In the storage layer, the protocol can ensure that 101 representative nodes can vote without interfering with each other, requiring a valid number, and everyone has only one chance to vote effectively. The CTF will also maintain a list of valid digital recipients in case someone attempts to change the ticket [30].

The selected RTN is selected as the complete node user who helps the thin client mining in the shared layer; the ATN supervises the mining process and detects dishonest nodes. If the existence of a dishonest node is detected, the highest sub-node in the ATN acts as an RTN, and an RTN node is re-elected in the mining pool. The detected dishonest nodes will no longer be eligible for election.

4.2.4. Full Node Working Process

In the EMRs shared layer, the RTN in the consortium blockchain acting as a full-node user helps the thin client to complete the working process to find the EMRs' index for the query, where the search process is completed using the K-anonymity algorithm. The sequence diagram of its authentication process is shown in Figure 6, and the authentication process is as follows:

1. **A → Patient**. Institution *A* sends an inquiry request to patient *P* and sends its identity ID_A and public key PK_A to the patient, waiting for the patient's authentication.
2. **Patient → RTN**. The system randomly selects $k-1$ IDs (such as *E*, *F*, ...), and sends IDA, IDE, IDF, ... a total of *k* IDs to these 30 full user nodes.
3. **RTN → Patient**. The 30 full user nodes traverse their own blockchain, and then find the corresponding public keys PK_A, PK_E, PK_F, ... of ID_A, ID_E, ID_F, ... and send these public keys to patients. This process uses the K-anonymity algorithm to hide the real information that needs to be found, thereby avoiding disclosure of patient *P* privacy in the presence of dishonest nodes.
4. **Patient → A**. If the PK_A received by the patient from *A* is the same as the PK_A received from the full-node user, then it can be determined that ID_A is from institution *A*. The patient then sends a message to *A* containing his own ID_P and a random number N_P, which is encrypted with the public key PK_A of institution *A*.
5. **A → Patient**. Institution *A* uses its private key SK_A to decrypt the received information to obtain ID_P and N_P, to find the public key PK_P corresponding to the patient. Then A uses PK_P to encrypt the information containing ID_A, N_A, N_P and sends it to the patient.
6. **Patient → A**. The patient decrypts the received information with his private key SK_P to obtain the N_A, and checks whether the N_P is in the message. If N_P is in the message, it means that institution *A* is the owner of ID_A, that is, the identity of institution *A* is legal. The patient then encrypts the information containing EMRs data, $\mathrm{sk}_{\mathrm{AES\text{-}CBC}}$ of AES-CBC and N_A and sends it to institution *A*, where the information is encrypted using $\mathrm{sk}_{\mathrm{AES\text{-}CBC}}$, PK_A.

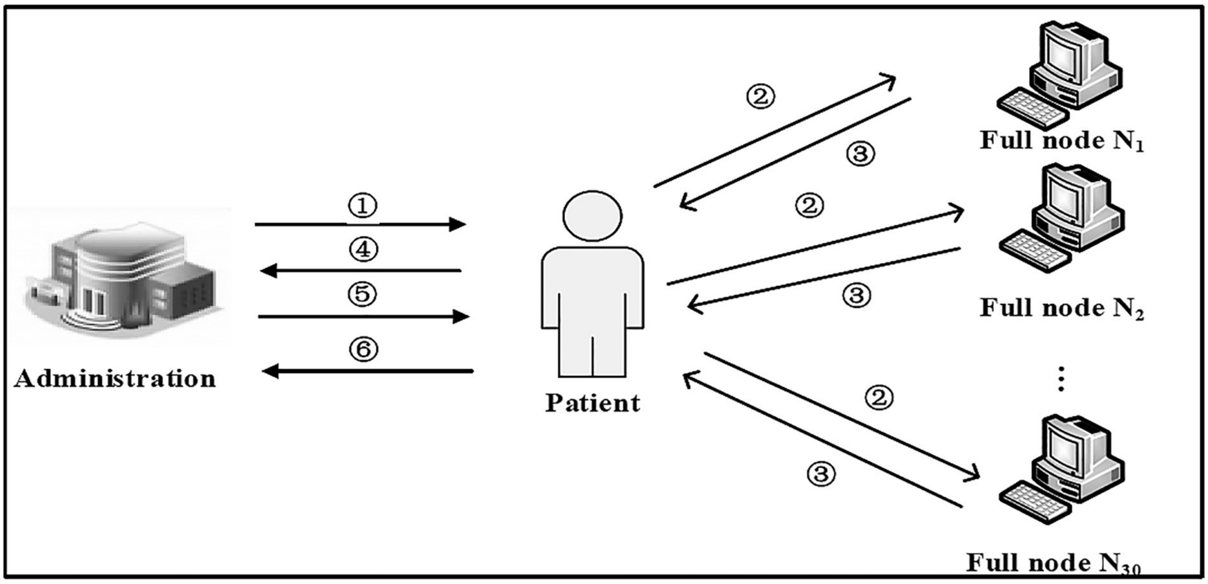

Figure 6. Implementation on Process of Thin-Clint.

7. Institution A uses its private key SK_A to decrypt the received information. If N_A is included in the received information, it is determined that the patient's identity information is true and valid. Institution A believes that the EMR data received is true and valid.

5. Security Analysis

5.1. Various Attacks and Security Precautions

In response to the current common blockchain attacks, this article makes corresponding security precautions for the consortium blockchain in the sharing model, as shown in Table 3.

Table 3. Common attack types and preventive measures in the consortium blockchain.

Attack type	Safety precautions
Witch attack	Malicious nodes cannot join the chain when the system starts.
Replay attack	Check the user identity sequence before each transaction.
51% attack	When the scale of the consortium chain is small, no service mechanism is provided.

1. For the problem of witch attacks, on the one hand, the system requires that nodes must be authenticated before they can join the blockchain platform. On the other hand, once the system is in normal operation, external nodes cannot join. Because the consortium chain is not like the public chain, not any node can join the chain and participate in decision-making, which can ensure that malicious nodes cannot join nodes as arbitrarily as the public chain, avoiding the occurrence of witch attacks.
2. The main feature of the replay attack is that the attacker hijacks the data packets that have been approved by the nodes on the chain and sends them again to achieve the purpose of destroying the blockchain data. To set a replay attack, when each node initiates a transaction, a globally increasing sequence number will be generated, and the sequence number will be verified each time to ensure that transactions written to the block never occur, thus ensuring data security.
3. Among blockchain attacks, 51% is currently the most serious attack. When the computing resources of a single node exceed those of other network nodes, this attack will occur, and then control the network to put malicious transactions into the blockchain. Although 51% attacks have not occurred in the Bitcoin network since the first block was created and added to the blockchain, its risk also exists, especially when the number of blockchain nodes is small. The most direct and effective way is to wait, that is, to set a threshold in the EMRs sharing model. When the number of blocks in the consortium blockchain exceeds this threshold, one lets it go for external service operations. This can effectively avoid the 51% attacks.

5.2. Dishonest Nodes

A dishonest node indicates that the node attempts to maliciously deceive the thin client. When a thin client asks a full-node user to query, selecting multiple nodes instead of one node can reduce the probability of being cheated. Patient P selects n complete user nodes. If there is only one dishonest node, patient P cannot be deceived. Only if all the n nodes are dishonest and colluding with each other, the patient P will be deceived.

Assuming that dishonest nodes account for c% of all complete nodes, the transaction results of honest nodes are consistent and the choice of full-node users is random, then we can draw the following conclusion: the probability of patient P being deceived is (c%)n. Assuming that the proportions of dishonest nodes in the system are 10%, 20%, and 30%, respectively, then the probability that patient P is deceived is related to n relationships as shown in Figure 7.

It can be seen from Figure 7 that when the probability of dishonest nodes in the system increases, the probability of thin clients being deceived will also increase accordingly. However, in the case of a certain proportion of dishonest nodes, the probability of thin clients being deceived will be reduced because n is worth increasing, where n is the number of thin clients selecting complete nodes. Therefore, we can conclude that in the consortium blockchain of the BPDST model, full nodes do not collude with each other, and thin clients are almost impossible to be deceived.

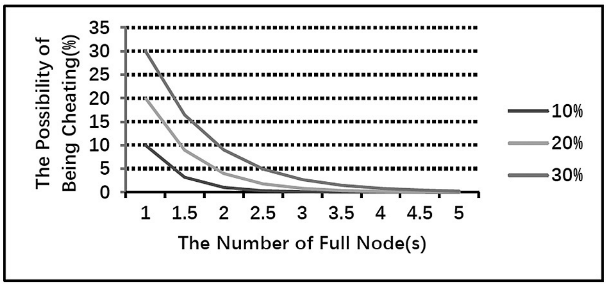

Figure 7. The probability of a thin client being deceived.

5.3. Performance Evaluation

We analyzed the calculation costs related to the BPDST approach.

• Calculation cost on a thin client: The calculation cost on a thin client refers to the time required for random number generation and related encryption and decryption operations, as shown in Table 4.

Table 4. The operation of implementing the process on a thin client.

Operation	Frequency
Generating a random number	2
Encryption	3
Decryption	3

We used Android Studio to test the average time of related operations on a thin client. The hardware parameters of the thin client were: CPU: Snapdragon756G, 2.4 GHz, Memory: 6 GB RAM. Each operation was executed 1000 times, and the average time of each operation finally obtained is shown in Table 5.

From the results of our simulation experiments, we can draw the following conclusion: the time cost of thin clients is within our acceptable range, which will not bring burden on the user of thin clients.

Table 5. The operations on a thin client.

Operation	Average time for the operation (ms)
Generating a random number	0.031
Encryption	0.384
Decryption	1.139

- Computational cost of full node users: This cost is determined by the number of searches performed by full-node users. *m*, *k* searches are required by the BPDST scheme. Figure 8 shows the number of searches required for the BPDST scheme, as *k* increases.

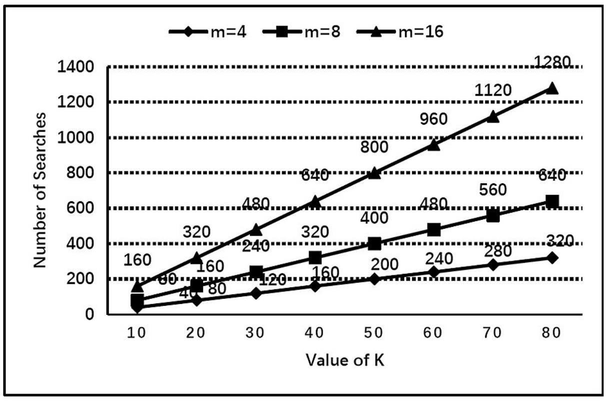

Figure 8. The number of required searches.

- Communication cost: We discuss the communication overhead between thin clients and full-node users. First, we set up that the length of ID is $d = 64$ bits, and the length of public key is $p = 1024$ bits. Figure 9 shows the total communication cost of BPDST, as *k* increases.

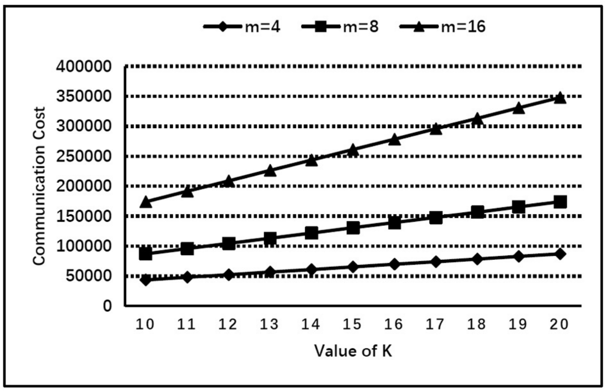

Figure 9. Communication cost of BPDST.

When storing EMRs, P_i will perform some encryption and decryption operations on the data of EMRs. As shown in Figure 10, the computation cost of P_i increases with the size of EMRs. The computation overhead of RSA increases with different sizes of data, as shown in Figure 11.

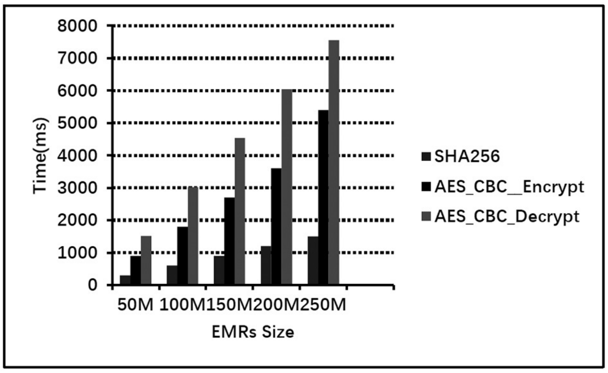

Figure 10. Computation overhead.

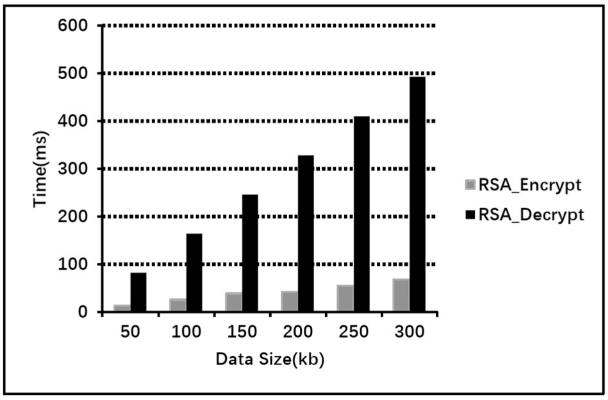

Figure 11. Computation overhead of RSA.

Based on the above simulation results, we can conclude that our scheme is feasible and protects the privacy of users.

6. Discussion

We deploy the relevant encryption and decryption algorithms on the thin client through Android Studio. The time for generating random numbers and performing encryption and decryption is relatively short each time. At the same time, when conducting AES_CBC on EMRs and related data, the encryption and decryption times of AES_CBC and RSA are also relatively short. The results show that, consistent with our expectations, it is feasible for us to share the privacy protection EMR data based on the blockchain on the thin client. Moreover, we have verified the data before and after encryption and decryption, and the results show that the data after encryption and decryption are consistent. Compared with literature [20], our consensus time has been shortened. Compared with literature [18], we have added K-anonymity and thin clients to better protect the privacy and convenience of patients. Therefore, our scheme effectively protects the EMRs' privacy of patients during data sharing, alleviates the storage pressure, provides convenience for users, and also breaks the "island" phenomenon among various institutions.

7. Conclusion

In this paper, we propose a sharing scheme based on blockchain privacy protection that can be executed on thin clients. For the characteristics of the thin clients and the blockchain, we have done relevant processing at each layer to get a shared model that can be operated on the thin clients. Through the BPDST approach proposed in this article, each patient can manage and safely share their own medical information among doctor, government, hospital and company through thin clients, without the risk of privacy information leakage. The security analysis shows that BPDST can achieve the security requirements. The experimental performance demonstrates the practicability and feasibility of BPDST.

Acknowledgment

This work was supported in part by the Yunnan Key Laboratory of Blockchain Application Technology under Grant 202105AG070005 (YNB202111), and in part by the National Natural Science Foundation of China under Grant62062047, Grant 61662042, and Grant 61962030.

References

[1] H.-Z. Sun *et al.*, "Impact of Electronic Medical Records Management Practices (Trial) on the Legal Effectiveness of Electronic Medical Records", *Chinese Hospital Management*, vol. 38, pp. 64–65, 2018.

[2] Y. Song *et al.*, "Research on Cloud Storage System Supporting Secure Sharing", *Journal on Communication*, vol. 38, pp. 88–96, 2017.

[3] C. Zhang *et al.*, "Medical Chain: Consortium Medical Blockchain System", *Acta Automatica Sinica*, vol. 45, pp. 1495–1510, 2019.

[4] G. Yu *et al.*, "The Challenge and Prospect of Distributed Data Management Techniques in Blockchain Systems", *Chinese Journal of Computers*, vol. 42, 2019.

[5] J. Liu *et al.*, "BPDS: A Blockchain Based Privacy-Preserving Data Sharing for Electronic Medical Records", in *Proc. of the 2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6. http://dx.doi.org/10.1109/GLOCOM.2018.8647713

[6] People's Daily, "The Number of Internet Users in China Reached 854 Million and the Proportion of Internet Users Using Mobile Phones Reached 99.1%", 2019.

[7] S. Nakamoto, "A Peer-to-Peer Electronic Cash System", 2019, available from: https://bitcoin.org/en/bitcoin-paper

[8] J. Kang *et al.*, "Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017. http://dx.doi.org/10.1109/TII.2017.2709784

[9] Y. Yuan *et al.*, "Blockchain Consensus Algorithms: The State of the Art and Future Trends", Acta Automatica Sinica, vol. 44, pp. 2011–2022, 2019.

[10] S. King and S. Nadal, "PPCoin: Peer-to-peer Crypto Currency with Proof-of-stake", 2012, available from: http://www.peercoin.net/

[11] D. Larimer, "DPOS Consensus Algorithm–The Missing White Paper", J. Bitshare whitepaper, 2017.

[12] D. Larimer, "Delegated Proof-of-stake (dpos)", J. Bitshare whitepaper, 2014.

[13] W.-Y. Xu, L. Wu and Y.-X. Yan, "A Medical Data Sharing Model via Blockchain", *Journal of Computer Research and Development*, vol. 55, pp. 2233–2243, 2019.

[14] H. Liu *et al.*, "Distributed K-Anonymity Location Privacy Protection Scheme Based on Blockchain", *Chinese Journal of Computers*, vol. 42, pp. 942–960, 2019.

[15] J. Fu, N. Wang, Y. Cai, "Privacy-Preserving in Healthcare Blockchain Systems Based on Lightweight Message Sharing", *Sensors (Basel)*, vol. 20, issue 7, p. 1898, 2020. http://dx.doi.org/10.3390/s20071898

[16] D.-S. Wang, "The Research on Privacy Protection for Outsourced Data in Cloud Computing", M. Sc. thesis, National University of Defense Technology, 2016.

[17] T.-F. Xue *et al.*, "A Medical Data Sharing Model via Blockchain", *Acta Automatica Sinica*, vol. 43, pp. 1555–1562, 2017. http://dx.doi.org/10.16383/j.aas.2017.c160661

[18] J. Liu *et al.*, "BPDS: A Blockchain Based Privacy-Preserving Data Sharing for Electronic Medical Records," in *Proc. of the 2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6. http://dx.doi.org/10.1109/GLOCOM.2018.8647713

[19] S. Alexaki *et al.*, "Blockchain-based Electronic Patient Records for Regulated Circular Healthcare Jurisdictions", in *Proc. of the 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018. http://dx.doi.org/10.1109/CAMAD.2018.8514954

[20] A. Azaria *et al.*, "MedRec: Using Blockchain for Medical Data Access and Permission Management", in *Proc. of the 2016 2nd International Conference on Open and Big Data (OBD)*, pp. 1–5, 2016. http://dx.doi.org/10.1109/OBD.2016.11

[21] K. N. Griggs *et al.*, "Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring", *Journal of Medical Systems*, vol. 42, pp. 130, 2018. http://dx.doi.org/10.1007/s10916-018-0982-x

[22] O. Gutiérrez *et al.*, "HealthyBlock: Blockchain-Based IT Architecture for Electronic Medical Records Resilient to Connectivity Failures", *International Journal of Environmental Research and Public Health*, vol. 17, p. 7132. http://dx.doi.org/10.3390/ijerph17197132

[23] S. Jayakumar *et al.*, "Blockchain Based Electronic Health Record System", *Social Science Electronic Publishing*, 2019. http://dx.doi.org/10.2139/ssrn.3761589

[24] J. Marquis *et al.*, "Blockchain Technology: Investing in a National EMR Strategy", *University of Western Ontario Medical Journal*, vol. 87, pp. 12–14, 2019. http://dx.doi.org/10.5206/uwomj.v87i2.1266

[25] L. Hang, E. Choi and D. H. Kim "A Novel EMR Integrity Management Based on a Medical Blockchain Platform in Hospital", *Electronics*, vol. 8, issue 4, p. 467, 2019. http://dx.doi.org/10.3390/electronics8040467

[26] S. G. Alonso *et al.*, "Proposing New Blockchain Challenges in eHealth", *Journal of Medical Systems*, vol. 43, issue 3, p. 64, 2019. http://dx.doi.org/10.1007/s10916-019-1195-7

[27] R. Johari and V. Kumar, "BLOSOM: Blockchain Technology for Security of Medical Records", *ICT Express*, vol. 8, issue 1, pp. 56–60, 2021. http://dx.doi.org/10.1016/j.icte.2021.06.002

[28] G. Wood, "Ethereum: A Secure Decentralized Generalised Transaction Ledger", 2018, [Online]. Available http://gavwood.com/paper.pdf

[29] Bitshares.org, Delegated proof-of-stake consensus, 2018, [Online]. Available https://bitshares.org/technology/delegatedproof-of-stake-consensus/

[30] Y.-K. Dong *et al.*, "Board Voting System based on the Consortium Blockchain", *Chinese Journal of Network and Information Security*, vol. 12, pp. 1–7, 2017.

Received: May 2022
Revised: September 2022
Accepted: October 2022

Contact addresses:

Lu Xu
Yunnan Provincial Key Laboratory of
Computer Technology Applications
Kunming University of Science and Technology
Kunming
China
e-mail: 1975212952@qq.com

Mengchen Lin
Yunnan Provincial Key Laboratory of
Computer Technology Applications
Kunming University of Science and Technology
Kunming
China
e-mail: lmc19940511@163.com

Yong Feng
Yunnan Provincial Key Laboratory of
Computer Technology Applications
Kunming University of Science and Technology
Kunming
China
e-mail: fybraver@163.com

Yani Sun
Yunnan Provincial Key Laboratory of
Blockchain Application Technology
Yunnan Innovation Research Institute of
Beijing University of Aeronautics and Astronautics
Kunming
China
e-mail: sunyani@bhynii.com

Lu Xu received the B.E. degree in software engineering from the Chongqing Normal University of China, Chongqing, in 2016. She is currently a graduate student with the Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming, China. Her current research interest is blockchain.

Mengchen Lin received a PhD degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2011. He is currently a Professor at the Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming, China. He has authored/co-authored three books and over 50 papers in peer-reviewed journals and conferences. His current research interests include Internet of Things, blockchain, and deep learning.

Yong Feng received the MSc degree from Kunming University of Science and Technology of China, Kunming, in 2021. His research interests are blockchain security and privacy protection.

Yani Sun received the MSc degree. She is currently an assistant researcher and office secretary of Yunnan Provincial Key Laboratory of Blockchain Application Technology of Yunnan Innovation Research Institute of Beijing University of Aeronautics and Astronautics of China. Her current research mainly includes the application of information management, information system and blockchain in government services, intelligent manufacturing, *etc.*