Future Generation Computer Systems 124 (2021) 338–350

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# A blockchain-based preserving and sharing system for medical data privacy

Zeng Chen, Weidong Xu, Bingtao Wang, Hua Yu *

*Key Laboratory of Optoelectronic Technology & Systems Ministry of Education, Department of Optoelectronic Engineering, Chongqing University, Chongqing 400044, China*
*International R & D Center of Micro-Nano Systems and New Materials Technology, Chongqing University, Chongqing 400044, China*

ARTICLE INFO

*Article history:*
Received 3 February 2021
Received in revised form 27 April 2021
Accepted 17 May 2021
Available online 27 May 2021

*Keywords:*
Blockchain
Hospital information system
Medical data collection and sharing
Proxy re-encryption algorithm
Hyperledger fabric

ABSTRACT

With the rapid development of information and network technology, hospital information systems (HIS) has also become a hot research area. However, data could be subjected to the threat of security attacks, leakage, tampering, and forgery during medical data transmissions, data storage, and sharing based on public networks and cloud environments. The immutability, decentralization and anonymity of the blockchain provided new ways for solving the aforementioned problems. This paper proposes a complete medical information system model based on blockchain technology, to realize the goal of safe storage and sharing of medical data. A data collection system based on Internet of Things (IoT) was also developed that can simultaneously collect data from different types of non-invasive medical instruments to realize real-time collection of patient health record during surgery (SHR). This system designed an anonymous medical data sharing scheme based on cloud servers and proxy re-encryption algorithm to improve the security of private medical data sharing. The system was implemented based on the permissioned blockchain architecture Hyperledger Fabric, and a dual-channel Fabric deployment architecture and medical chaincode were designed for data management and access control. We also carried out computing overhead test, performance test and safety evaluation on the system, and the test results meet the requirements of actual medical production environment. The research work of this paper provides means for remote diagnosis and treatment, data mining and other practical applications based on the medical data on the blockchain.

## 1. Introduction

Hospital information system (HIS) is the infrastructure of modern hospitals. It plays a vital role in numerous aspects such as patient management, diagnosis and treatment decision. HIS manages the hospital with modern, scientific, and standardized means, which can effectively improve the hospital's work efficiency and medical quality [1]. The patient's medical data contains the patient's historical physiological information, which has important reference value for the patient's disease diagnosis and treatment, and daily health care [2]. Li et al. [3] summarized the common medical data types as electronic health records (EHR), electronic medical records (EMR) and personal health records (PHR). The patient's medical data is the basis for the normal operation of the HIS, so the HIS usually collects, stores and manages various types of medical data of the patient. However, some problems exist in traditional HIS: (1) System security issues and difficulties in medical data sharing. Traditional HIS stores medical data in private or cloud servers, which exposes the system to the risk of malicious attacks, data leakage and tampering, and the possibility of losing private medical data. Besides, these systems usually operate independently in hospitals, which is difficult to meet the basic need of patients to share medical data. Some patients will share medical data through some cloud servers or applications, but cloud servers provided by third-party suppliers are semi-trusted and they may steal or tamper with user data. (2) Lack of patient surgical data. The physiological data of patients during surgery generated by medical instruments were not included in the medical data types summarized by Li et al. [3]. These patient health records during surgery (SHR) reflect the patient's surgical process and facilitates the postoperative treatment and recovery of patients. The traditional method of manually recording SHR has the problems of limited storage and data collection inaccuracy. (3) Lack of analysis and application of medical data. The medical data in traditional HIS is solely managed by hospital and patients cannot access their own medical data at will, which prevents patients from effectively analyzing and using their own

* Corresponding author at: Key Laboratory of Optoelectronic Technology & Systems Ministry of Education, Department of Optoelectronic Engineering, Chongqing University, Chongqing 400044, China.
*E-mail address:* yuhua@cqu.edu.cn (H. Yu).

https://doi.org/10.1016/j.future.2021.05.023
0167-739X/© 2021 Elsevier B.V. All rights reserved.

data. Most of the medical data is idle, resulting in a waste of valuable data resources. Internet of Things has been widely used in many fields [4]. Its real time and accuracy make it very suitable for medical data collection [5]. Blockchain is a distributed digital ledger based on encryption technologies [6], which is represented as a decentralized peer-to-peer network. Once data is confirmed and stored in the blockchain, it is difficult to tamper with. The immutability, decentralization and anonymity of the blockchain ensure the safe storage of medical data. The combination of IoT and blockchain technology provides a new way for solving the above problems.

Blockchain technology had been applied to various fields, and medical information system is among the main focuses at present [7–10]. Kassab et al. [11] and Abu-Elezz et al. [12] have conducted research on common data protection mechanisms and analyzed the advantages of blockchain in the security protection and sharing of medical data. There have been many studies on introducing blockchain technology into electronic medical record systems, medical image systems, etc. [13–17]. The systems proposed in [13–17] only use blockchain for secure medical data storage. They focused on architectural design but lacked of system implementation details and data sharing procedures. Some researchers proposed schemas or frameworks for medical data storage based on blockchain and various cryptographic mechanisms [18–21]. They designed specific system workflows to realize the sharing or access control of medical data. However, the proposed systems were implemented based on Ethereum, even normal transactions in these systems require a certain number of tokens, which is not applicable in hospital information systems. Besides, Xu et al. [22] adopted the PBFT consensus algorithm and LIU et al. [23] adopted the DPOS consensus algorithm in their systems, which were relatively complex for reaching a systematic consensus and ensure the security of the data. The systems proposed in [18–23] have the weakness of low transaction efficiency, because traditional public blockchain architectures such as Bitcoin and Ethereum need to use high-power or relatively complex algorithms to reach a system consensus. Therefore, some researchers had adopted the more lightweight Hyperledger Fabric(HF) architecture for blockchain network implementation in their systems [24–27]. They had developed smart contracts for system access control and data management but had less research on data sharing. Proxy re-encryption algorithm had been applied to different scenarios to achieve secure data sharing on semi-trusted cloud servers [28], but its application in the medical system based on blockchain was relatively insufficient. The Internet of Things technology has been applied to improve the efficiency of medical data collection generated by the medical instruments or personal health monitoring sensors [29–34]. But the focus of these studies are on the construction of data collection system or data analysis. There is a lack of research on the secure storage and sharing of medical data. Many systems also analyzed and mined medical data and make contributions to disease prediction and medical quality improvement [35–37].

Taking into account the problems of traditional HIS and the current research status, this paper proposed a medical data information system model based on blockchain, Internet of Things, cloud storage and proxy re-encryption algorithm to realize the reliable collection, safe storage and sharing of medical data. The contributions and innovations of this work are as follows.

① A Hyperledger Fabric based medical data information system with dual channels and medical chaincode was proposed to realize the goal of safe storage, management and access control of medical data.

② An IoT based medical data system was designed to realize real-time data collection for six categories of non-invasive medical instruments, which can effectively improve the data collection efficiency of SHR.

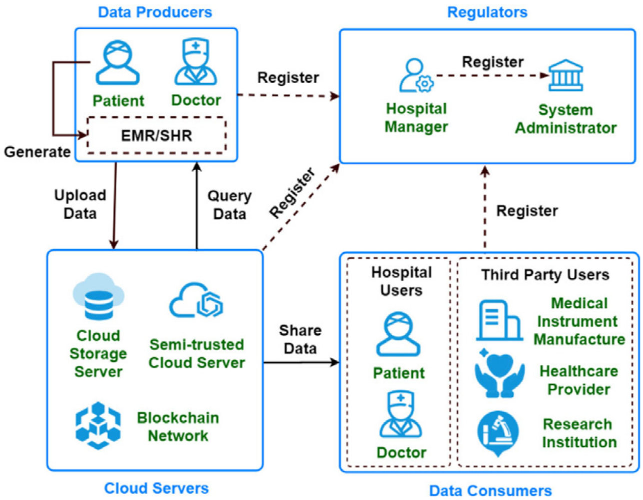


**Fig. 1.** System participants.

③ A secure medical data sharing scheme based on proxy re-encryption algorithm was proposed to improve the security of medical data sharing.

## 2. System model design

### 2.1. Architecture design of the proposed model

#### 2.1.1. System design for participants

In order to meet the regulatory requirements of government medical institutions on medical systems and comply with the organization and management of traditional hospitals, this system was designed to include a system administrator group (SA) composed of government medical institutions and managers of hospitals (HM). SA supervises the system and issues digital certificates to legitimate registered users who have passed identity authentication. HM manages doctors and patients in the hospital. The other main participants of the system are shown in Fig. 1.

① **Data producers**: Data producers include doctors and patients. Doctors diagnose and treat patients, thus generating EMR/SHR data. The patient can upload the data to the cloud server and blockchain network for storage or retrieve the data and share it with other users.

② **Cloud servers**: Cloud storage servers are used to store encrypted medical data. The index record of medical data such as data storage location or digital signature is stored in the blockchain. Semi-trusted cloud servers provide blockchain access APIs and proxy services for secret key conversion during data sharing process.

③ **Data consumers**: In addition to hospital users such as doctors and patients, data consumers also include third-party users such as medical instrument manufacturers, healthcare providers and research institutions.

#### 2.1.2. Structure and workflow design of the proposed system

In order to be consistent with the organization and management of traditional hospitals and improve the transaction efficiency of HIS, the system was designed and implemented based on the permissioned blockchain architecture Hyperledger Fabric. The architecture and workflow design of the system are shown in Fig. 2.

The system was designed into five layers according to the functional structure. (1) System management layer. This layer includes SA from government medical institutions and HM of

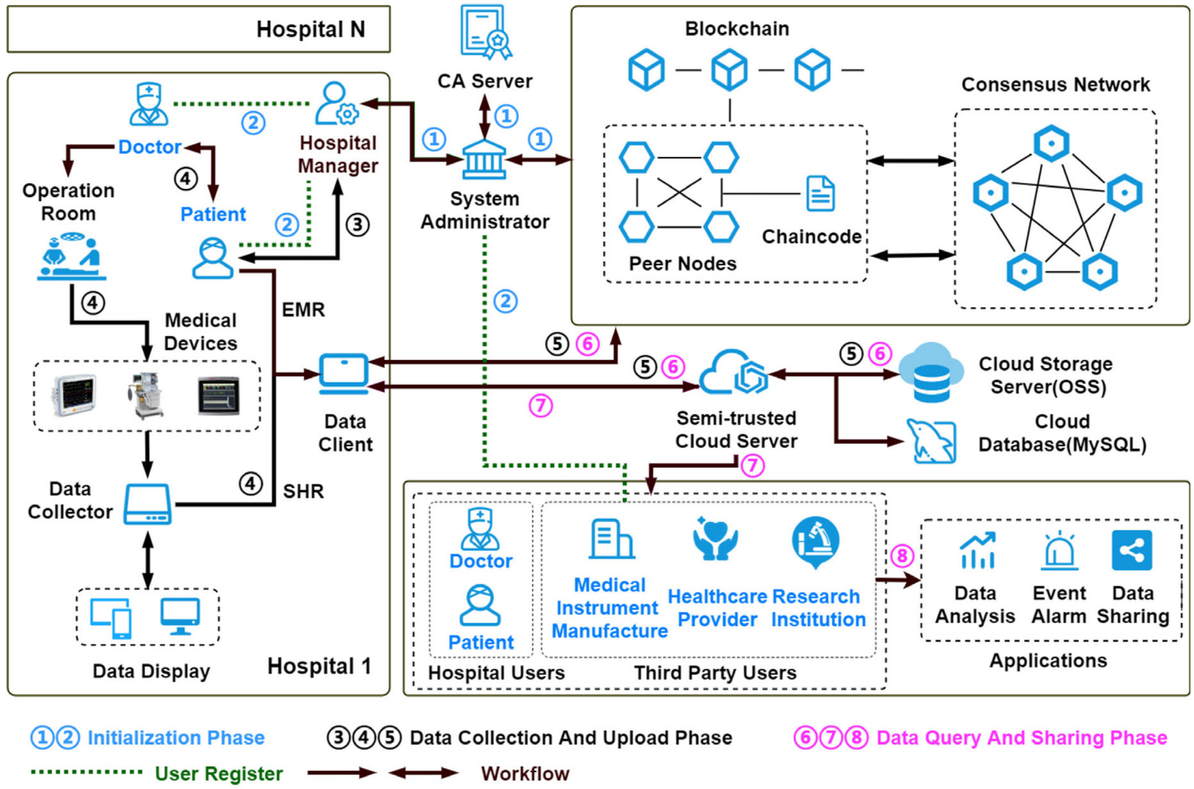


**Fig. 2.** Architecture and workflow of the system.

each hospital. They are mainly responsible for user registration, identity authentication, certificate issuance, and system supervision. The system certificate authority servers (CA) are managed by SA. (2) Data collection layer. EMR and SHR data can be collected with the help of doctors and data collector in this layer. The collected medical data are sent to the data client application for processing and uploading. (3) Blockchain network layer. This layer consists of the consensus network, server nodes provided by hospitals and third parties, data ledger, and chaincode. Medical data index records and data usage records are stored in the blockchain. Specific chaincodes were designed for ledger data management and user access control. (4) Cloud service layer. This layer is composed of semi-trusted cloud server, cloud storage server and cloud database. RESTFul interfaces of cloud server were designed for interactions of data clients, blockchain access, data management, and data sharing. (5) Application layer. In this layer, patients can share historical medical data with data consumers for various applications.

The workflow of the system was designed according to the basic steps of system initialization, data collection and upload, data querying and sharing, as shown in Fig. 2. The detailed design and explanation are as follows.

### 2.2. Specific design of system workflow

This section covers the specific design of the system workflow based on the proposed model. The symbols and operations used in this paper are shown in Tables A.1 and A.2.

#### 2.2.1. Initialization phase

The aim of the initialization phase is to complete the system network construction and user registration. In order to achieve secure data storage and communication, users need to provide identity information for registration, and then obtain TLS certificate and X.509 certificate. TLS certificate enables users to communicate safely in the system through TLS protocol. X.509 is the identity certificate of server nodes and users, which stores the public key of the user.

**① System network construction phase**

Government medical institution (GMI) organize and build the blockchain network based on Hyperledger Fabric. Firstly, GMI configures the encryption parameters of the system and initiates the CA Servers. Then GMI designates the system administrator (SA), who then initializes Orderer, Zookeeper and Kafka server nodes in the blockchain network through configuration files ($CF$).

$$SetUp_{SA(CF)}(Orderer \parallel Zookeeper \parallel Kafka)$$

The hospital $H_j$ and the third-party organization $TH_j$ that join the system shall provide identity information to SA for qualification audit. TLS and X509 certificates will be generated for them after the approval by SA. The $H_j$ and $TH_j$ are supposed to provide and setup peer node servers in the blockchain network.

$$KeyGen_{(SA\to x)}(PK_x \parallel SK_x),\ x \in (H_j/TH_j)$$
$$CertGen_{(SA\to x)}(Cert_{tls}(x) \parallel Cert_{x509}\ (x)),\ x \in (H_j/TH_j)$$
$$SetUp_{x(Cert_{tls}(x),Cert_{x509}(x),CF)}(Peer),\ x \in (H_j/TH_j)$$

SA creates channels for the system, then adds peer nodes to the corresponding channel, and installs and instantiates the chaincode on each peer node.

$$SA \xrightarrow{Cert(x)} Join\ Channel\ (Peer)$$
$$\xrightarrow{Cert(x)} Install\ CC\ (Peer) \xrightarrow{Cert(x)} Instantiate\ CC\ (Peer)\ ,$$
$$x \in (H_j/TH_j),\ Cert(x) \in (Cert_{tls}(x), Cert_{x509}(x))$$

After the network is created, the hospital $H_j$ needs to initialize its hospital manager $HM_j$ and register with SA. SA will generate key pairs, TLS and X.509 certificate for $HM_j$.

$$KeyGen_{(SA\rightarrow HM_j)}(PK_{HM_j} \parallel SK_{HM_j})$$
$$CertGen_{(SA\rightarrow HM_j)}(Cert_{tls}(HM_j) \parallel Cert_{x509}(HM_j))$$

New hospitals or third-party institutions joining the system should also follow the aforementioned procedure.

**② User registration phase**

All users are supposed to be registered to exclude the system from irrelevant personnel and to ensure safe data communication. The system will generate digital certificates and pseudo identities for the registered users. Doctors and patients are the producers and consumers of patient data, while the third-party institutional users are just data consumers. In order to facilitate user management, patients and doctors are registered and managed by hospital administrators, while the third-party institutions are registered and managed by SA.

The user $U_x$ joining the system needs to generate a key pair and prepare his real identity information (RID) before registration.

$$KeyGen_{(U_x\rightarrow U_x)}(SK_x \parallel PK_x), x \in (D_{i,j}, P_{i,j}, TH_{i,j})$$

The doctor $D_{i,j}$ or patient $P_{i,j}$who registers in the hospital $H_j$ need to submit a registration request and provide their real identity information (RID) and public key $PK_x(x \in (D_{i,j}, P_{i,j}))$ to hospital manager $HM_j$. The request will be forwarded to the SA for verification, while users of third-party institutions should directly provide the above information to SA.

$$D_{i,j}/P_{i,j} \xrightarrow{RID_x\|PK_x} HM_j \xrightarrow{RID_x\|PK_x} SA, x \in (D_{i,j}, P_{i,j})$$
$$TH_{i,j} \xrightarrow{RID_{TH_{i,j}}\|PK_{TH_{i,j}}} SA$$

SA will grant X.509 certificates and TLS certificates to users who pass the registration audit. Meanwhile, SA uses the user's public key to encrypt the user's RID and calculates its message digest to generate the system pseudo identity for the user.

$$CertGen_{(SA\rightarrow x)}(Cert_{tls}(x) \parallel Cert_{x509}(x))$$
$$H_{SHA384}(En_{PK_x(RSA)}(RID_x)) \rightarrow PID_x, x \in (D_{i,j}, P_{i,j}, TH_{i,j})$$

SA then stores user's RID and PID information to the private database and stores the X.509 certificate into the cloud database. Finally, SA returns certificate and PID to the user who has successfully registered. System users can obtain digital certificates from cloud database through the API provided by cloud servers.

### 2.2.2. Data collection and upload phase

The data collection and uploading stage is divided into several sub-stages with reference to the medical treatment process of traditional hospitals.

**③ Patient appointment phase**

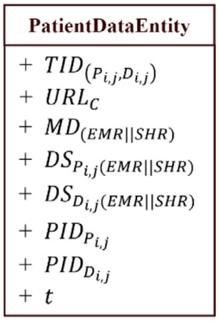
In this phase, the registered patient provides his health information to the HM, who then will assign a doctor and generate the appointment information for him. When patient $P_{i,j}$ needs medical treatment from hospital $H_j$, the patient is asked to submit an appointment request to $HM_j$. The request carries the *PID*, the digital certificate and the disease information (*DI*) of the patient.

$$P_{i,j} \xrightarrow{(PID_{P_{i,j}}\|Cert_{x509}(P_{i,j})\|DI_{P_{i,j}})} HM_j$$

$HM_j$ then generates a treatment ID ($TID_{(P_{i,j},D_{i,j})}$) and assigns a doctor $D_{i,j}$ to the patient who has received the appointment certification.

$$HM_j \xrightarrow{verify(PID_{P_{i,j}}\|Cert_{x509}(P_{i,j}))} TID_{(P_{i,j},D_{i,j})}$$

Then the $HM_j$ uses the public key of the doctor to encrypt the pseudo identity, the assigned $TID_{(P_{i,j},D_{i,j})}$, the disease information, and the digital certificate of the patient to generate a treatment session key $TSK_{(P_{i,j},D_{i,j})}$.


PatientDataEntity

+ $TID_{(P_{i,j},D_{i,j})}$
+ $URL_C$
+ $MD_{(EMR||SHR)}$
+ $DS_{P_{i,j}(EMR||SHR)}$
+ $DS_{D_{i,j}(EMR||SHR)}$
+ $PID_{P_{i,j}}$
+ $PID_{D_{i,j}}$
+ $t$

**Fig. 3.** Data structure of PatientDataEntity.

$$En_{PK_{D_{i,j}}(RSA)}(PID_{P_{i,j}} \parallel TID_{(P_{i,j},D_{i,j})} \parallel DI_{P_{i,j}} \parallel Cert_{x509}(P_{i,j}))$$
$$\rightarrow TSK_{(P_{i,j},D_{i,j})}$$

Finally, $HM_j$ will return the $TSK_{(P_{i,j},D_{i,j})}$ and necessary treatment information such as time, place and doctor's name to the patient.

**④ Treatment and data collection phase**

This phase completes the generation and collection of medical data. The patient carries the $TSK_{(P_{i,j},D_{i,j})}$ to the designated doctor for treatment. $D_{i,j}$ will use his private key to decrypt the $TSK_{(P_{i,j},D_{i,j})}$ and access the information storing in it.

$$De_{SK_{D_{i,j}}(RSA)}(TSK_{(P_{i,j},D_{i,j})})$$
$$\rightarrow (PID_{P_{i,j}} \parallel TID_{(P_{i,j},D_{i,j})} \parallel DI_{P_{i,j}} \parallel Cert_{x509}(P_{i,j}))$$

Then the doctor diagnoses and treats the patient according to the disease information or historical medical data, and EMR data ($M_{EMR}$) is generated for the patient. If the patient needs surgery, doctor would use the data collector to extract the patient's SHR data ($M_{SHR}$) generated by medical instruments. Both EMR and SHR will be gathered to the data client application and then confirmed by doctor and patient. After confirmation, the patient encrypts the data with the public key to get the ciphertext of the medical data, and then applies SHA384 hash algorithm to calculate message digest of the encrypted medical data.

$$En_{PK_{P_{i,j}}(RSA)}(M_{EMR} \parallel M_{SHR}) \rightarrow C_{(EMR\|SHR)}$$
$$H_{SHA384}(C_{(EMR\|SHR)}) \rightarrow MD_{(EMR\|SHR)}$$

The patient uses the private key to sign the message digest and pseudo identity to obtain the digital signature $DS_{D_{i,j}(EMR\|SHR)}$ of the encrypted medical data. The doctor also performs the same operation to obtain the $DS_{P_{i,j}(EMR\|SHR)}$. In this way, data users can query index record of medical data from the blockchain and verify digital signatures.

$$En_{SK_{P_{i,j}}(RSA)}(MD_{(EMR\|SHR)} \parallel PID_{P_{i,j}}) \rightarrow DS_{P_{i,j}(EMR\|SHR)}$$
$$En_{SK_{D_{i,j}}(RSA)}(MD_{(EMR\|SHR)} \parallel PID_{D_{i,j}}) \rightarrow DS_{D_{i,j}(EMR\|SHR)}$$

Patient will get $C_{(EMR\|SHR)}$, $MD_{(EMR\|SHR)}$, $DS_{P_{i,j}(EMR\|SHR)}$, $DS_{D_{i,j}(EMR\|SHR)}$ and $TID_{(P_{i,j},D_{i,j})}$ after this phase.

**⑤ Data upload and storage phase**

After treatment and data collection phase, the encrypted medical data is uploaded to the cloud storage server and the data storage address $URL_C$ can be obtained. Then the patient can construct the data information obtained in the above process into the data structure PatientDataEntity as shown in Fig. 3, thereby forming the index record of the medical data.

After that, the patient uses the data client application to convert the PatientDataEntity into a JSON string and upload it to

**Table 1**
Save index record of medical data to blockchain.

```
Algorithm 1 Save index record of medical data to blockchain
Input: (1) entity: An object of PatientDataEntity that contains index record
of medical data. (2) cc: The chaincode going to be called. (3) bcn:
Architecture information of the blockchain network. (4) client: A HFClient
client agent provided by Fabric SDK, which is initialized by $PK_{P_{i,j}}$ and $SK_{P_{i,j}}$.
Output: true/false
Function saveIndexData(entity, cc, bcn, client):
If(!entity. isValid()) then:
  Return false;
String data = toJSONString(entity);
List<ProposalResponse> proResList = new List();
For(EndorseNode node in bcn.endorseNodes):
  ProposalResponse res = client.sendProposal(node, cc, args(data));
  proResList.add(res);
For(ProposalResponse res in proResList):
  If(!res.isValid()) then:
    Return false;
List<TransactionResponse> transResList =
client.sendTransaction(bcn.ordererNodes, proResList);
For(TransactionResponse res in transResList):
If(!res.isValid()) then:
  Return false;
Return true;
```

the peer node representing each hospital, to execute the corresponding chaincode for endorsement. If the endorsement comes through, it will finally be submitted to the Orderer node to preserve the data in the blockchain network. The process of using data client to store the index records of medical data in the blockchain is shown in Algorithm 1 (see Table 1). The data client is an application developed based on Fabric SDK. The HFClient provided by Hyperledger Fabric SDK is required for blockchain access and data management.

### 2.2.3. Data query and sharing phase

The system has designed a complete data sharing schema to tackle the difficulties of data sharing and limited data usage applications in traditional HIS.

**⑥ Historical data query phase**

The patient who shares data with others should first obtain his historical medical data. The patient can call the chaincode in the blockchain to query the medical data index record and fetch the PatientDataEntity object according to the pseudo identity and historical treatment information. After that, the patient can retrieve the encrypted data storage address $URL_C$ from the PatientDataEntity and obtain the corresponding encrypted medical data from the cloud storage server.

$$P_{i,j} \xrightarrow{GET:URL_C} C_{(EMR\|SHR)}$$

After obtaining the encrypted data, the patient calculates its message digest and verifies the digital signature in the index record obtained from the blockchain to check whether the data has been tampered with. The verification process is as follows.

$$H_{SHA384}(C_{(EMR\|SHR)}) \rightarrow MD_{(EMR\|SHR)}$$
$$verifySign_{PK_{P_{i,j}}(RSA)}(MD_{(EMR\|SHR)}, DS_{P_{i,j}(EMR\|SHR)}) \rightarrow True/False$$

If the signature is verified, the patient uses his private key to decrypt the encrypted data to obtain the plaintext of the medical data.

$$De_{SK_{P_{i,j}}(RSA)}(C_{(EMR\|SHR)}) \rightarrow (M_{EMR} \| M_{SHR})$$

**⑦ Data sharing phase**

In order to solve the problems of data theft or tampering that may exist in the cloud server during data sharing, the system adopted a proxy re-encryption algorithm to ensure the safe sharing of data. The semi-trusted cloud server only provides proxy services for secret key conversion and cannot obtain any plaintext data.

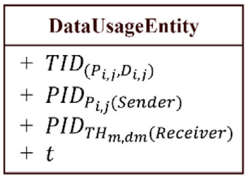

DataUsageEntity
+ $TID_{(P_{i,j},D_{i,j})}$
+ $PID_{P_{i,j}(Sender)}$
+ $PID_{TH_{m,dm}(Receiver)}$
+ $t$

**Fig. 4.** Data structure of DataUsageEntity.

A data sharing request that contains the digital certificate and pseudo identity will be generated by the data consumer once he or she needs to use the patient medical data. The patient can verify the consumer certificate through SA to determine whether the recipient is a legitimate system user.

$$TH_{U_x} \xrightarrow{(Cert_{x509}(TH_{U_x}),PID_{TH_{U_x}})} P_{i,j}$$

We assume that a certain instrument manufacturer (Data Receiver) uses the medical data of the patient (Data Sender). Firstly, the patient randomly generates a symmetric encryption key $DSK_{AES^P}$ based on the AES algorithm, and then the patient uses the key to encrypt the plaintext of medical data and part of the information contained in the index record to obtain the shared data ciphertext $C_{s(EMR\|SHR)}$.

$$KeyGen_{(P_{i,j}\rightarrow P_{i,j})}(DSK_{AES^P})$$
$$En_{DSK_{AES^P}(AES)}((M_{EHR} \| M_{SHR}) \| PID_{P_{i,j}} \| TID_{(P_{i,j},D_{i,j})}) \rightarrow C_{s(EMR\|SHR)}$$

After the encryption is finished, the patient applies the RSA algorithm and his public key to encrypt the original AES key $DSK_{AES^P}$ to generate the encrypted shared key $DSK_{AES^{P*}}$.

$$En_{PK_{P_{i,j}}(RSA)}(DSK_{AES^P}) \rightarrow DSK_{AES^{P*}}$$

Then the patient applies the RSA algorithm and the public key of the data receiver to encrypt its secret key to generate a proxy re-encryption conversion key $RK_{P\rightarrow TH}$. And $C_{s(EMR\|SHR)}$, $DSK_{AES^{P*}}$, $RK_{P\rightarrow TH}$, $PID_x(x \in (P_{i,j}, TH_{m,dm}))$, and $Cert_{x509}(x \in (P_{i,j}, TH_{m,dm}))$ will be sent to the semi-trusted cloud server.

$$En_{PK_{TH_{m,dm}}(RSA)}(SK_{P_{i,j}}) \rightarrow RK_{P\rightarrow TH}$$

The semi-trusted cloud server verifies the received certificates and the pseudo identity of the users through the chaincode. If the data sender and receiver are both legitimate registered users, the proxy re-encryption algorithm will be employed to convert the conversion key and generate the encrypted AES key $DSK_{AES^{TH*}}$ for the consumer.

$$PKeyReGen(RK_{P\rightarrow TH}, DSK_{AES^{P*}}) \rightarrow DSK_{AES^{TH*}}$$

The cloud server sends the converted encryption key $DSK_{AES^{TH*}}$ and encrypted data to the data receiver $TH_{m,dm}$. After that the semi-trusted cloud server constructs the medical data usage record into the data structure DataUsageEntity as shown in Fig. 4. The server converts the DataUsageEntity into a JSON string and uploads it to the blockchain network for persistent storage. Users can then obtain the data usage record by querying the blockchain to achieve data supervision.

After receiving data from the cloud server, $TH_{m,dm}$ uses the private key to decrypt the conversion key to obtain the original AES key $DSK_{AES^{TH}}$. The plaintext of the medical data, the pseudo identity and treatment information of the patient can be obtained by decrypting the encrypted data with the AES key.

$$De_{SK_{TH_{m,dm}}(RSA)}(DSK_{AES^{TH*}}) \rightarrow DSK_{AES^{TH}}$$

$$De_{DSK_{AES}TH\,(AES)}(C_{s(EMR\|SHR)}) \rightarrow (M_{EMR} \parallel M_{SHR}) \parallel PID_{P_{i,j}} \parallel TID_{(P_{i,j},D_{i,j})}$$

The data receiver can also verify the received historical data to ensure it has not been maliciously tampered with by patients or other actors. The data receiver can use the patient's public key to encrypt the received data plaintext, and then apply the same hash algorithm as the one used by the patient to calculate the message digest of the data.

$$H_{SHA384}(En_{PK_{P_{i,j}}(RSA)}(M_{EMR} \parallel M_{SHR})) \rightarrow MD_{(EMR\|SHR)}$$

The receiver then queries the corresponding index record of the medical data in the blockchain according to the received $PID_{P_{i,j}}$ and $TID_{(P_{i,j},D_{i,j})}$, which contains the patient's digital signature for the data. Then the receiver uses the patient's public key, the message digest, and the $PID_{P_{i,j}}$ to verify the signature, thereby judging whether the received data has been tampered with.

$$verifySign_{PK_{P_{i,j}}(RSA)}((MD_{(EMR\|SHR)} \parallel PID_{P_{i,j}}), DS_{P_{i,j}(EMR\|SHR)})$$
$$\rightarrow True/False$$

⑧ **Data usage phase**

This stage is a functional expansion of traditional hospital information system. The system users can process and use the patient's historical medical data to implement various applications. Patients can communicate about their conditions through the system based on historical medical data, and doctors can use historical medical data generated in other hospitals, thus achieving inter-hospital data sharing. Medical data is also of great significance for third-party organizations. For example, scientific research institutions and medical service providers can do some data mining based on medical data to obtain more exhaustive information. Medical instrument manufacturers can also evaluate the performance of the instruments through the patient's SHR, which then can be used to guide instruments refinements and upgrades.

## 3. Implementation of the proposed system

### 3.1. Implementation of the data collection system and cloud service layer

In order to solve the OHR data recording problem in traditional HIS, this system had developed a data collection system based on the Internet of Things to realize real-time collection of physiological data generated by various non-invasive medical monitoring instruments. The architecture design of the data collection system is shown in Fig. 5(a).

The data collector was designed based on STM32 microcontroller to realize medical instrument communication, data collection and parsing. The embedded control program was custom-designed according to the specific communication and parsing protocol of medical instruments. The data collector integrates RS232 and RJ45 interfaces to ensure the compatibility with different interface types of medical instruments. The multi-interface design of the data collector makes possible the data collection of 6 different types of medical instruments, which enabled it to collect more exhaustive medical data than traditional HIS. The collector also designed a communication module based on WiFi and 4G. The WiFi module can transmit data to the mobile pad in the operating room. A specially designed and developed Android APP program is installed on the pad to realize the comprehensive display of various types of medical data, which improved the efficiency and convenience of doctors checking the data of multiple instruments during surgery. The 4G module can transmit the collected data to the data client for processing and uploading. The developed data collector and its working status in the operating room are shown in Figs. 5(b), 5(c) and 5(d).

This system had designed different RESTFul interfaces for users, which were implemented based on the Spring framework. Users can communicate with the server based on the HTTP protocol to achieve user management, blockchain access, data sharing, etc. The cloud server was deployed in combination with each peer node, and the distributed deployment method can be adopted to achieve high system availability and reduce the impact of single point of failure on the entire system.

### 3.2. Implementation of the blockchain network layer

#### 3.2.1. Deployment structure and organization design

This system was implemented based on the Hyperledger Fabric blockchain network, which is represented as a distributed network composed of multiple different types of server nodes. In order to break the information barriers between traditional HIS, this system maps the entities of different hospitals and third-party organizations as peer nodes in the network. The architecture design diagram of the blockchain network is shown in Fig. 6.

In order to improve transaction efficiency, this system adopted a consensus mechanism based on Kafka, which is more lightweight and efficient than consensus algorithms such as PoW and PBFT in other blockchain architectures. A consensus network composed of Orderer nodes, Zookeeper cluster and Kafka cluster were deployed to achieve the stability and reliability of the sorting service.

The system users were divided into the patient organization (OrgPatient) and the third-party organization (OrgThirdParty) according to their roles and relationships with others. The patient organization includes all doctors and patients. Each hospital in the patient organization needs to provide a server for deploying peer node and maintain the ledger of medical data index record. The third-party organizations are composed of third-party institutions other than hospitals, including government medical institutions and data consumers such as medical instrument manufacturers, research institutions and healthcare providers. All actors in third-party organizations can provide multiple peer server nodes. For example, the government medical institutions can deploy several peer nodes to prevent untrustworthy supervision.

The system also set up several CA server nodes managed by SA to implement user certificate issuance and key pair generation. Each of the patient organization and the third-party organization has an intermediate CA node server, which is endorsed by the Root CA, thus forming a certificate chain. The identity verification of users in the system is based on cascading verification.

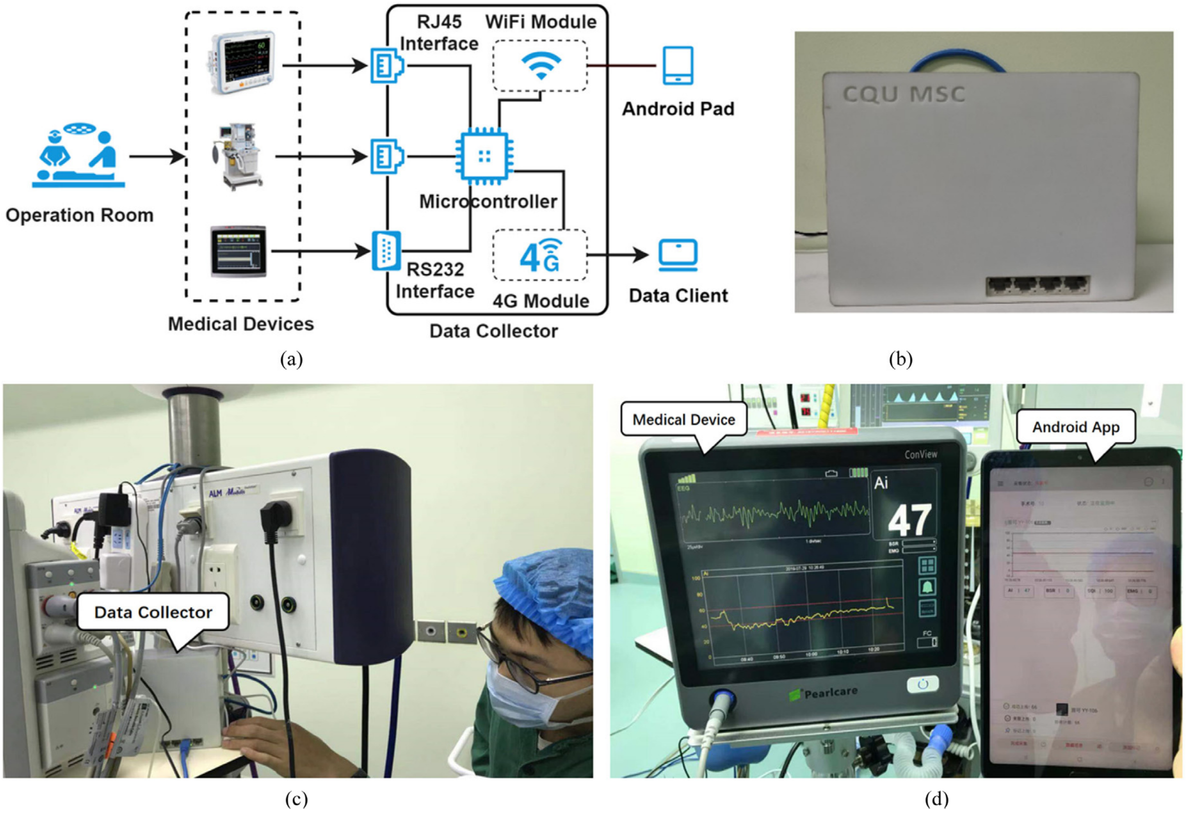
#### 3.2.2. Channel and chaincode design

Many third-party organizations only use the data and will not participate in the generation and collection of patient medical data.

In order to isolate the index record of patient medical data from third-party institutions, a dual-channel architecture that consists of Patient Channel (PC) and Data Usage Channel (DUC) was designed to improve the storage security of medical data. The patient channel contains the peer nodes managed by the hospitals and government medical institutions, and the index record of medical data will be maintained. The DUC contains all peer nodes in the system, which stores the usage record information of medical data. It is worth noting that the peer nodes of government medical institutions need to join both channels and two data ledgers should be maintained to meet the demand of system supervision.

① **Patient channel**

All peer nodes in the PC jointly maintain the index record ledger and store the index information of the medical data generated during the patient's treatment and operation. The key



(a)

(b)

(c)

(d)

**Fig. 5.** (a) Architecture design of the data collection system. (b) The data collector. (c) The installed data collector in the operation room. (d) A medical device and the Android APP for data display.

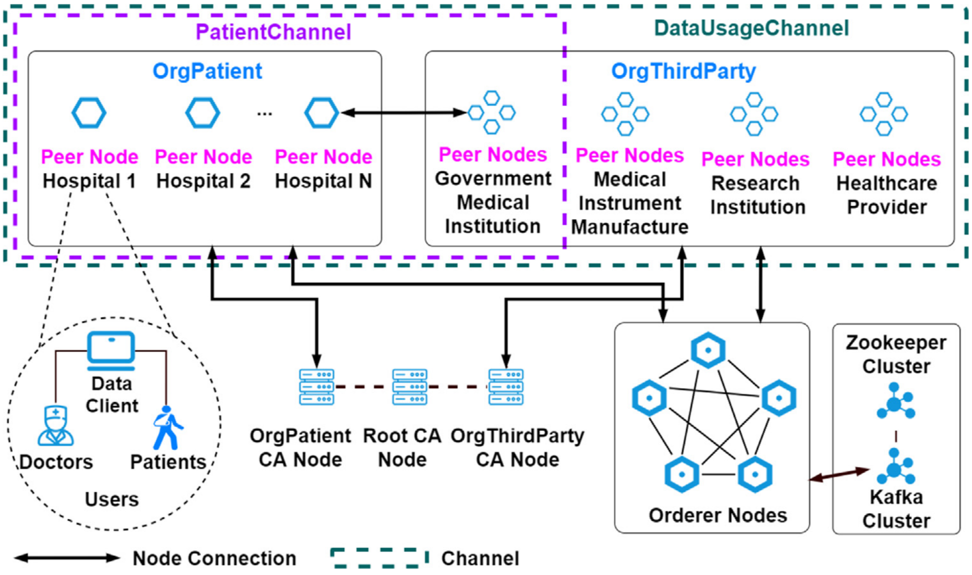


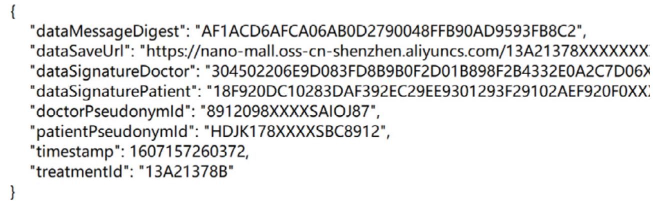
**Fig. 6.** Deployment architecture of the blockchain network.

in the ledger is the patient's pseudo identity, and the value is the JSON string of the PatientDataEntity structure corresponding to the index record of medical data. A sample JSON string of PatientDataEntity is shown in Fig. 7.

This system has designed chaincode with different methods for PC to realize the storage, access and verification of the index record, as shown in Algorithm 2 (see Table 2).

② **Data usage channel**

The data usage channel that contains all the nodes will jointly maintain the data usage record ledger to store the usage record of patient data. When a patient shares data with data consumers through proxy re-encryption algorithm, the semi-trusted cloud server will construct a data usage record and upload it to the DUC, so that the users and the government medical institutions can track the data usage information. The key in the data usage ledger is the patient's pseudo identity, and the value is the JSON string of the DataUsageEntity structure composed of data usage record. In order to facilitate subsequent inquiries, both the data sender and the receiver need to store the same data usage record. A sample JSON string of DataUsageEntity is shown in Fig. 8.



```
{
  "dataMessageDigest": "AF1ACD6AFCA06AB0D2790048FFB90AD9593FB8C2",
  "dataSaveUrl": "https://nano-mall.oss-cn-shenzhen.aliyuncs.com/13A21378XXXXXXX
  "dataSignatureDoctor": "304502206E9D083FD8B9B0F2D01B898F2B4332E0A2C7D06X
  "dataSignaturePatient": "18F920DC10283DAF392EC29EE9301293F29102AEF920F0XX
  "doctorPseudonymId": "8912098XXXXSAIOJ87",
  "patientPseudonymId": "HDJK178XXXXSBC8912",
  "timestamp": 1607157260372,
  "treatmentId": "13A21378B"
}
```

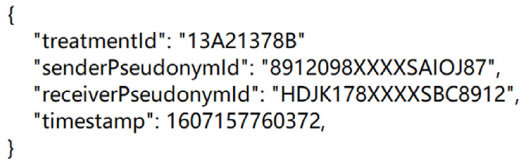
**Fig. 7.** Sample JSON data of PatientDataEntity.

**Table 2**
Chaincode of patient channel.

**Algorithm 2** Chaincode of patient channel

**Input:** **(1) enStr**: The JSON string of a PatientDataEntity object. **(2) ledger**: The object provided by the chaincode API to access the blockchain ledger. **(3) pid:** $PID_{P_{i,j}}$ of $P_{i,j}$. **(4) tTid**: The target $TID_{(P_{i,j},D_{i,j})}$ of the medical data index record being queried. **(5) md**: Message digest of the data ready for verification.
**Output**: Result of the calling function.

```
Function Init():
  Initialize the chaincode.
Function savePatientData(enStr, ledger):
  PatientDataEntity entity = JSONStringToObject(enStr);
  String data = ledger.getState(entity.pid);
  List<PatientDataEntity> dataList =
  JSONStringToObjectList(data);
  dataList.add(entity);
  String newData = objectListToJSONString(dataList);
  ledger.putState(entity.pid, newData);
  Return success;
Function queryByPidAndTid (pid, tTid, ledger):
  String data = ledger.getState(pid);
  If(data == null) then:
    Return null;
  List<PatientDataEntity> dataList =
    JSONStringToObjectList(data);
  For(PatientDataEntity entity : dataList):
    If(entity.tid == tTid) then:
      Return entity;
  Return null;
```



```
{
  "treatmentId": "13A21378B"
  "senderPseudonymId": "8912098XXXXSAIOJ87",
  "receiverPseudonymId": "HDJK178XXXXSBC8912",
  "timestamp": 1607157760372,
}
```

**Fig. 8.** Sample JSON data of DataUsageEntity.

Algorithm 3 is the chaincode algorithm for managing medical data usage record (see Table 3).

**Table 3**
Chaincode of data usage channel.

**Algorithm 3** Chaincode of data usage channel

**Input:** (1) **enStr**: The JSON string of a DataUsageEntity object. (2) **ledger**: The object provided by the chaincode API to access the blockchain ledger. (3) **pid:** Pseudo identity of a user $U_x, x \in (P_{i,j}, D_{i,j}, TH_{i,j})$. (4) **tTid**: The target $TID_{U_x}$ of the data usage record being queried.
**Output**: Result of calling the function.

```
Function Init():
  Initialize the chaincode.
Function saveDataUsageData(enStr, ledger):
  DataUsageEntity entity = JSONStringToObject(enStr);
  String senderData = ledger.getState(entity.senderPid);
  List<DataUsageEntity> senderList =
    JSONStringToObjectList(senderData);
  senderList.add(entity);
  String newData = objectListToJSONString(senderList);
  ledger.putState(entity.senderPid, newData);
  String receiverData = ledger.getState(entity.receiverPid);
  List<DataUsageEntity> receiverList =
    JSONStringToObjectList(receiverData);
  receiverList.add(entity);
  newData = objectListToJSONString(receiverList);
  ledger.putState(entity.receiverPid, newData);
  Return success;
Function queryByPidAndTid (pid, tTid):
  String data = ledger.getState(pid);
  If(data == null) then:
    Return null;
  List<DataUsageEntity> dataList =
    JSONStringToObjectList(data);
  For(DataUsageEntity entity : dataList):
    If(entity. tid == tTid) then:
      Return entity;
  Return null;
```

## 4. System evaluation

### 4.1. Performance evaluation

The performance test of the system was carried out to evaluate the efficiency of medical data processing and sharing. In order to simulate the production environment, network construction and business development were implemented according to the design in the previous chapters. A total of seven virtual machines were deployed in the test. The installation relationships between different types of server nodes and virtual machines are shown in Table A.3. And the software and hardware environment are shown in Tables A.4 and A.5. Apache JMeter is the performance testing tool used.

We have carried out computing overhead test to evaluate the system performance. We ignored the testing of one-time computing overhead phases, such as the system construction and patient registration phase. The annotations of the tested operations are shown in Table A.6. We used different sizes of unencrypted medical data files (64KB, 256Kb and 1MB) to test each operation for 1000 times. The average values of the test results are shown in Table 4. Furthermore, the computing overhead of different phases can be calculated, as shown in Table 5.

The test results shows that the computing overhead of $t^*_{er}$, $t^*_{dr}$, $t^*_{md}$, $t^*_{uo}$, $t^*_{do}$, $t^*_{ea}$ and $t^*_{da}$ increase with the size of the unencrypted medical data file. However, these are calculated and completed by the user's personal data client applications, thus do not affect the performance of the blockchain system or semi-trusted cloud server. The test results also show that the computing overhead of operating blockchain network has no obvious relationship with the size of the medical data file. Besides, the computing overhead is high and reaches several seconds, thus the efficiency of the blockchain network is crucial to the performance of the entire system. Therefore, we also tested the throughput of the blockchain network to evaluate the system performance in the production environment.

When the hardware and software environments are determined, the performance of the blockchain network will be affected by parameters such as consensus protocol, the preferred maximum block size (BS), and the maximum message count (MC) in a block. The Solo and Kafka consensus protocols were supported in the tested Fabric version. In order to simulate the production environment, we chose the Kafka consensus protocol and enabled the TLS protocol to ensure safe data transmission. In addition, the batch timeout of block generation was set to 2 s. The method to calculate the request rate (*RPS*) is as follows.

$$Request\ Per\ Second\ (RPS) = \frac{(Total\ Thread\ Number)}{(Ramp\ Up\ Time)}$$

The parameter Ramp-Up time of JMeter was set to a fixed 60 s in the test, thus different RPS can be obtained by continuously

**Table 4**
Test results of computing overhead for important operations of the system (in ms).

| File size | $t^*_{er}$ | $t^*_{dr}$ | $t^*_{md}$ | $t^*_{uo}$ | $t^*_{do}$ | $t^*_{sig}$ | $t^*_{vs}$ | $t^*_{ea}$ | $t^*_{da}$ | $t^*_{kg}$ | $t_{pkrg}$ | $t_{ap}$ | $t_{qp}$ | $t_{ad}$ | $t_{qd}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **64 KB** | 396 | 2115 | 3 | 392 | 351 | 5 | 1 | 2 | 3 | 2 | 3 | 2150 | 27 | 2089 | 43 |
| **256 KB** | 1590 | 8453 | 13 | 495 | 479 | 4 | 1 | 6 | 7 | 2 | 3 | 2145 | 26 | 2148 | 40 |
| **1 MB** | 6348 | 35707 | 45 | 1403 | 1354 | 4 | 1 | 20 | 26 | 2 | 3 | 2144 | 24 | 2027 | 39 |

**Table 5**
The computing overhead of different phases (in ms)

| Operation Time Notations(ms) | Computation overhead | Calculation results<br>(File size 64 KB/256 KB/1 MB) |
|---|---|---|
| Treatment and data collection phase | $t^*_{er} + t^*_{md} + 2t^*_{sig}$ | 409/1611/6401 |
| Data upload and storage phase | $t^*_{uo} + t_{ap}$ | 2542/2640/3547 |
| Historical data query phase | $t_{qp} + t^*_{do} + t^*_{md} + t^*_{vs} + t^*_{dr}$ | 2497/8972/37131 |
| Data sharing phase | $t^*_{kg} + t^*_{ea} + t_{pkrg} + t_{ad} + t^*_{da} + t^*_{md} + t_{qp} + t^*_{vs}$ | 2130/2206/2148 |

**Table 6**
Parameters of performance test.

| Item | Message count | Block size (KB) |
|---|---|---|
| Add index record of medical data | 5,10,15,30,75 | 512 |
| | 100 | 32,64,128,256 |
| Share patient data | 100 | 512 |
| Query index record of medical data | 100 | 512 |
| Query data usage record | 100 | 512 |

adjusting the total thread number. Besides, we kept increasing the number of threads during each test until the error rate is higher than 3%.

The performance test method of the Fabric blockchain network in [38] was referenced in this paper. We carried out the throughput tests on the performance of adding and querying index record of patient data, sharing patient data, and querying data usage records. The parameters of each test are shown in Table 6.

In order to facilitate system performance test, we developed several HTTP interfaces for blockchain access and data (a) (b) sharing[1]. When testing the interface of adding medical data index record, we set one of MC and BS to a fixed larger value, and continuously increases the value of another parameter to measure the performance of the system under different parameters. The test results are shown in Figs. 9(a) and 9(b). The results show that as MC and BS increase within a certain range, the maximum throughput of the system will also increase. However, after exceeding a threshold, the maximum throughput tends to stabilize to 140, which can be considered as reaching the maximum load of the system. Based on the above results, a fixed MC value of 100 and a BS value of 512KB will be set in subsequent tests, which ensures that the performance of the system will not be restricted by the consensus protocol parameters.

This paper also tested the QPS of medical data sharing process based on proxy re-encryption algorithm. The sharing process involved in the test includes user key pair generation, encryption of medical data by data sender, proxy re-encryption through cloud server, data usage record uploading to DUC, and decryption of medical data by data receiver. The test result is shown in Fig. 9(c). The results show that the maximum throughput of the system for data sharing is still around 140, which is similar to that of adding patient data index record.

We also carried out the tests on the performance of querying medical data index record (MDIR) and data usage record (DUR). The test method was still to keep the values of MC and BS unchanged, and to increase the number of request threads per second continually until the response error rate was greater than 3%. The test result is shown in Fig. 9(d). It can be seen that the maximum throughput of obtaining MDIR and DUR information are almost the same at about 166. Although the average response time (ART) delay of querying MDIR is about 10 ms lower than querying DUR information, the average response delays of both are less than 100 ms. This indicates that the data query efficiency of this system is relatively high. The time-consuming endorsement operation needs to be executed when adding data, but not when querying. This will result in a lower efficiency of adding data, which is consistent with the test results.

The test throughput of the system was slightly lower than the result of [38]. The reason is that [38] used a simple and efficient Solo consensus protocol, while this system used the Kafka consensus protocol, which is less efficient but more stable and convenient for system expansion. Moreover, we enabled TLS transmission to realize secure data transmission, which also has a certain degree of impact on the performance of the system. In general, the test environment and parameter settings in this paper are more realistic for production environments and hold greater significance for realization of the project.

### 4.2. Security evaluation

The security analysis of the proposed system is as follows.

① Data transmission security. All users and server nodes in the system passed identity authentications and have the TLS and digital certificates. The communication and data transmission in the system are based on the TLS protocol, which ensures the security of data transmission.

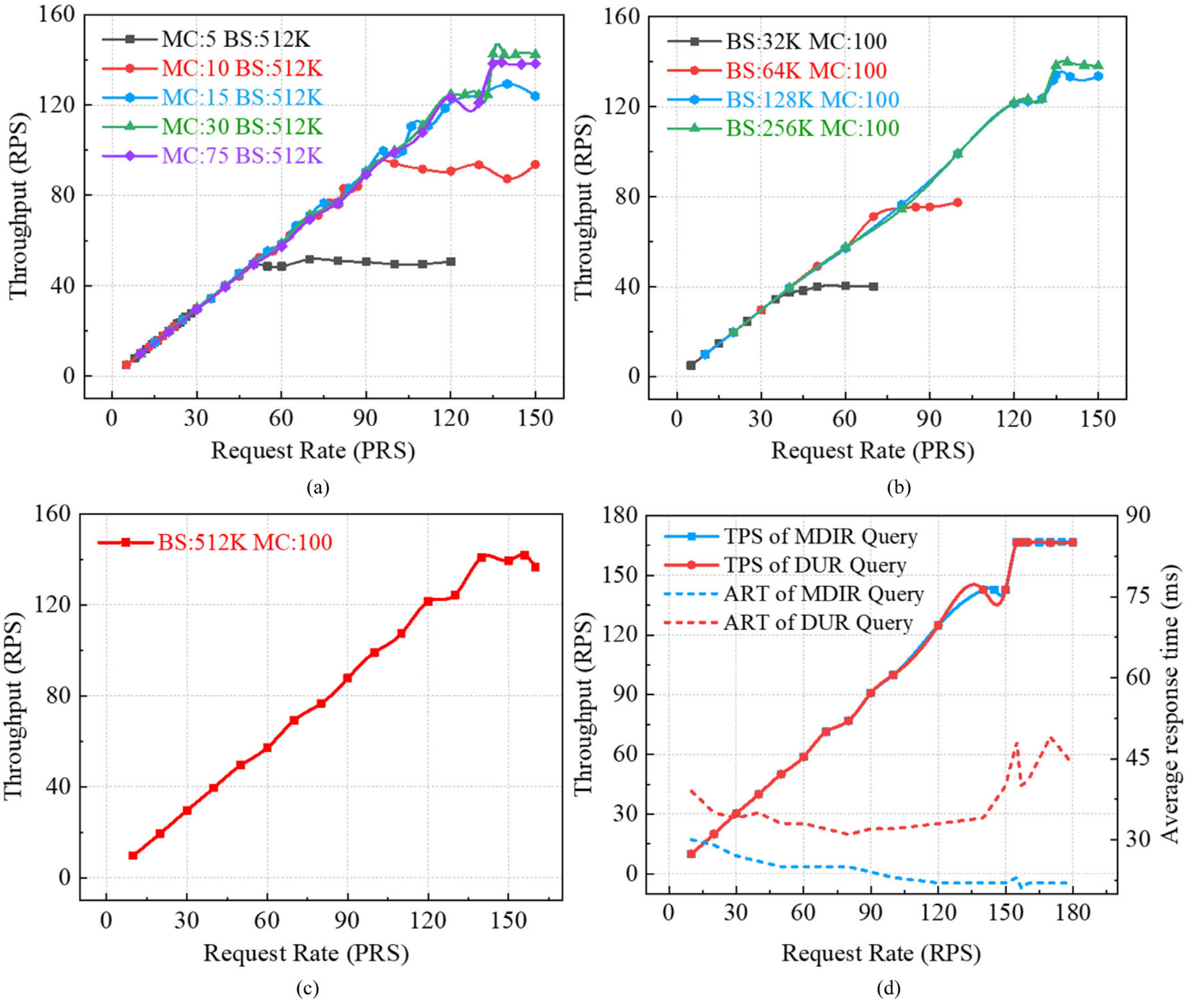
② User anonymity. All users in the system perform data management and sharing based on pseudo identities, so that only data recipients trusted by the patient can access the pseudo-identity information of each other, while others cannot. The user's real identity information will not be transmitted in plaintext through public network. Attackers cannot track the identity nor collect the medical data of a specific user.

③ Replay Attack. Replay attack is common in blockchain systems. The prevention of replay attacks in this system depends on the endorsement, sorting and verification mechanism of Hyperledger Fabric, which is simple and efficient.

④ Data storage security. The encrypted medical data in this system are stored on semi-trusted cloud storage servers provided by third-party cloud service providers, and the data storage security is guaranteed by the cloud service providers. Users need to perform signature verification when querying or sharing data to ensure that the encrypted data has not been tampered with.

The functionality and security comparison among the system and other similar systems or schemes are shown in Table 7. Compared with similar systems, the proposed system has the

[1] GitHub: https://github.com/nanodaemony/MedicalLedger



(a) (b) (c) (d)

**Fig. 9.** Results of system performance test. (a) (b) Test result of add index record of medical data. (c) Test result of data sharing. (d) Test result of querying MDIR and DUR.

**Table 7**
Comparison of this system with other systems or schemes.

| Items | [22] | [20] | [23] | [24] | [27] | Our system |
|---|---|---|---|---|---|---|
| No tokens | × | × | ✓ | ✓ | ✓ | ✓ |
| Data sharing | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| IoT implemented | ✓ | × | × | × | ✓ | ✓ |
| Contains SHR | × | × | × | × | × | ✓ |
| User anonymity | × | ✓ | ✓ | × | × | ✓ |

merits of a detailed data collection and sharing process. It also has certain advantages in transaction processing efficiency and the compatibility with different types of collected data.

## 5. Conclusion

In order to solve the problems of traditional medical information system, this paper proposed a blockchain-based medical data information system to realize the secure collection, storage and sharing of EMR and SHR data. A IoT based medical data collection system was proposed to collect the data generated by medical instruments during surgery, which can improve the efficiency and reliability of SHR data collection. The blockchain based data storage system and the proxy re-encryption algorithm based medical data sharing scheme improved the security of medical data storage and sharing, and ensured the safe sharing of medical data. Therefore, data consumers can employ the shared data in numerous applications, such as disease communication, inter-hospital treatment, and data mining. The computing overhead of important operations in the workflow was tested in our proposed system. The test result shows that the efficiency of the blockchain network is crucial to the performance of the entire system. Furthermore, we carried out the performance test of the blockchain network in a production environment. The test results showed that the throughput of data adding in this system is about 140, and the throughput of data querying is about 166. The safety evaluation showed that this system is safer than traditional HIS. In general, we have designed a complete information model that includes the collection, storage, query and sharing of medical data, and has created a research foundation for building a safer and more efficient medical information system. The focus of our follow-up research is to analyze and mine medical data based on this system.

**Table A.1**
The notations in this paper.

| Notation | Description |
|---|---|
| $SA$ | System administrator |
| $CF$ | Configuration files |
| $H_j$ | Hospital $j$ in the system |
| $HM_j$ | Manager of the hospital $j$ |
| $D_{i,j}$ | Doctor $i$ working in the hospital $j$ |
| $P_{i,j}$ | Patient $i$ in the hospital $j$ |
| $TH_j$ | The third-party organizations in system, $j \in (dm, ri, mp)$, here $dm$ is medical instrument manufacturers, $ri$ is research institutions, $mp$ is medical healthcare providers |
| $TH_{i,j}$ | A user $i$ in a third-party organization $TH_j$ |
| $U_x$ | A user in the system, $x \in \left(D_{i,j}, P_{i,j}, TH_{i,j}\right)$ |
| $RID_i$ | Real identity of user $i$ |
| $PID_i$ | Pseudo identity of user $i$ |
| $MD_{(x)}$ | Message digest of data $x$ |
| $DS_{i(x)}$ | Digital signature of data $x$, which is signed by user $i$ |
| $DI_{P_{i,j}}$ | Disease information of patient $P_{i,j}$ |
| $TID_{(P_{i,j},D_{i,j})}$ | Treatment ID that indicates $P_{i,j}$ is getting treatment from $D_{i,j}$ |
| $TSK_{(P_{i,j},D_{i,j})}$ | Treatment session key binding with TID |
| $Cert_t(U)$ | Certificate file $t$ belonging to user $U$, $t \in (x509, tls)$, here $x509$ is the X.509 certificate file and $tls$ is the TLS certificate |
| $M_{(x)}, C_{(x)}$ | M refers to plaintext and C refers to ciphertext |
| $CA$ | Certificate Authority |
| $EMR, SHR$ | $EMR$ refers to electronic medical record, $SHR$ refers to surgical health record |
| $PK_i, SK_i$ | Public key and secret key of user $i$ |
| $SHA384$ | $SHA384$ secure hash algorithm |
| $DSK_{AES^{U*}}$ | A data sharing key base on symmetric encryption algorithm AES for user $U$, here $*$ indicates that this key had been encrypted |
| $RK_{P\rightarrow TH}$ | A proxy re-encryption conversion key, which is generated by $P$ and will be used to convert the $DSK_{AES}$ for $TH$ |
| $\|\|$ | Connect two adjacent messages |

**Table A.2**
The operations in this paper.

| Notation | Description |
|---|---|
| $SetUp_{U(SM)}(x)$ | User $U$ starts and initialize a server node $x$ in the blockchain network by supporting materials $SM$, $SM \in (Cert_t\ (U)\ , CF)$ |
| $KeyGen_{(g\rightarrow u)}(x)$ | User $g$ generates key or key pairs $x(x \in (PK_i, SK_i, DSK_{AES}))$ for user $u$ |
| $CertGen_{(g\rightarrow u)}(Cert_t(U))$ | User $g$ generates certificate $Cert_t(U)$ for user $u$ |
| $JoinChannel\,(Peer)$ | Add a Fabric peer node to a channel |
| $InstallCC\ (Peer)$ | Install the specified chaincode on a Fabric peer node |
| $InstantiateCC(Peer)$ | Instantiate the specified chaincode on a Fabric peer node |
| $Sign_{k(SA)}(x)$ | Apply the secret $k$ and the signature algorithm $SA$ to sign the content $x$ and get the signed data |
| $verify(x)$ | Verify whether content $x$ is valid or not, $x \in (PID_i, Cert_t(U))$ |
| $verifySign_{k(SA)}(x, DS_x)$ | Verify the information $x$ and its digital signature $DS_x$ with the secret $k$ and signature algorithm $SA$ |
| $En_{k(A)}(M)$ | Use the public key $k$ and encryption algorithm $A$ to encrypt the plaintext $M$ to get the encrypted data |
| $De_{k(A)}(C)$ | Use the secret key $k$ and decryption algorithm $A$ to decrypt the ciphertext $C$ to obtain the decrypted data. |
| $H_A(x)$ | Use the specified hash algorithm $A$ to calculate the message digest of content $x$ |
| $GET : URL_S$ | Obtain resources from the specified URL by HTTP GET method |
| $PKeyReGen(RK_s, DSK_{AES^s})$ | Apply conversion key $RK_s$, $DSK_{AES^s}$ from data sender, and proxy re-encryption algorithm to generate $DSK_{AES^r}$ for data receiver |

**Table A.3**
Correspondence between virtual machine and server nodes installation.

| IP Address | Peer | Orderer | CouchDB | Zookeeper | Kafka | CA |
|---|---|---|---|---|---|---|
| 172.20.29.30 | peerxinqiao$^p$ | orderer0 | couchdb0 | zookeeper0 | kafka0 | |
| 172.20.29.31 | peerkunyi$^p$ | orderer1 | couchdb1 | zookeeper1 | kafka1 | ca0 |
| 172.20.29.32 | peerhuaxi$^p$ | orderer2 | couchdb2 | zookeeper2 | kafka2 | |
| 172.20.29.33 | peergovernment$^t$ | | couchdb3 | | kafka3 | ca1 |
| 172.20.29.34 | peermindray$^t$ | | couchdb4 | | | |
| 172.20.29.35 | peercqu$^t$ | | couchdb5 | | | |
| 172.20.29.36 | peerhealthcare$^t$ | | couchdb6 | | | |

Here peer node (x)$^p$ belongs to the patient organization and (x)$^t$ belongs to the third-party organization.

**Table A.4**
Hardware environment of the performance test.

| Item | Specifications |
|---|---|
| Host CPU | 24 CPUs x Intel(R) Xeon(R) Gold 6136 CPU @ 3.00 GHz |
| VM CPU | Intel(R) Xeon(R) Gold 6136 CPU @ 3.00 GHz |
| VM Memory | 8 GB |
| VM Disk | 20 GB |
| Android Tablet | 4 GB(Memory)/64 GB(SD Card)/ Snapdragon 660AIE(CPU) @ 2.2 GHz |

**Table A.5**
Software environment of the performance test.

| Item | Specifications |
|---|---|
| VM Host OS | Ubuntu 18.04 |
| Hyperledger Fabric(Peer/Order/CA) | V1.4.0 |
| Spring Boot | V2.2.0 |
| Docker-Compose | V1.24.0-rc1 |
| Docker Engine | V19.03.13 |
| Fabric-Zookeeper | V1.4.0 |
| Fabric-Kafka | V1.4.0 |
| CouchDB | V1.4.0 |
| Android Version | 8.1.0 |

**Table A.6**
The computational notations in this paper.

| Operation time (ms) | Description |
|---|---|
| $t^{*}_{er}$ | Encrypting data with RSA algorithm |
| $t^{*}_{dr}$ | Decrypting data with RSA algorithm |
| $t^{*}_{md}$ | Apply SHA384 algorithm to calculate the message digest |
| $t^{*}_{sig}$ | User signs the data and obtains the digital signature |
| $t_{ap}$ | Construct a PatientDataEntity object and upload it to the PC |
| $t_{qp}$ | Query PatientDataEntity data from the PC |
| $t^{*}_{ea}$ | Encrypting data with AES algorithm |
| $t^{*}_{da}$ | Decrypting data with AES algorithm |
| $t^{*}_{kg}$ | Generate the conversion key and the data sharing key |
| $t_{pkrg}$ | Apply proxy re-encryption algorithm and data sharing key to convert the data sharing key |
| $t_{ad}$ | Construct a DataUsageEntity object and upload it to the DUC |
| $t_{qd}$ | Query DataUsageEntity data from the DUC |
| $t^{*}_{uo}$ | Upload the encrypted data file to the cloud storage server(OSS) |
| $t^{*}_{do}$ | Download the encrypted data file from the cloud storage server(OSS) |
| $t^{*}_{vs}$ | Verify the digital signature and the data |

Here operations with a "*" is performed on the data client application.

## CRediT authorship contribution statement

**Zeng Chen:** Conceptualization, Methodology, Software, Resources, Writing - original draft. **Weidong Xu:** Data curation, Formal analysis. **Bingtao Wang:** Validation, Investigation. **Hua Yu:** Supervision, Writing - review & editing, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The research was supported by the National Key Research and Development Program of China (Grant No. 2018YFC0117202), the National Nature Science Foundation of China (Grant No. 61874015), and the Chongqing Venture & Innovation Support Program for Chongqing Overseas Returnees, China.

## Appendix

See Tables A.1–A.6.

## References

[1] S.Y. Sun, Z. Xie, K.T. Yu, B.Q. Jiang, S.W. Zheng, X.T. Pan, COVID-19 and healthcare system in China: Challenges and progression for a sustainable future, Glob. Health 17 (2021) 1–8.

[2] D.W. Bates, S. Saria, L. Ohno-Machado, A. Shah, G. Escobar, Big data in health care: Using analytics to identify and manage high-risk and high-cost patients, Health Aff. (Millwood) 33 (2014) 1123–1131.

[3] C.T. Li, D.H. Shih, C.C. Wang, C.L. Chen, C.C. Lee, A blockchain based data aggregation and group authentication scheme for electronic medical system, IEEE Access 8 (2020) 173904-173917.

[4] H. Liu, H. Ning, Q. Mu, Y. Zheng, J. Zeng, L.T. Yang, R. Huang, J. Ma, A review of the smart world, Future Gener. Comput. Syst. 96 (2019) 678–691.

[5] C.A. Tokognon, B. Gao, G.Y. Tian, Y. Yan, Structural health monitoring framework based on internet of things: A survey, IEEE Internet Things J. 4 (2017) 619–635.

[6] T.F. Lee, H.Z. Li, Y.P. Hsieh, A blockchain-based medical data preservation scheme for telecare medical information systems, Int. J. Inf. Secur. (2020) 1–13.

[7] M.A. Khan, K. Salah, IoT Security: Review blockchain solutions open challenges, Future Gener. Comput. Syst. 82 (2018) 395–411.

[8] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, M. Peng, When Internet of Things meets blockchain: Challenges in distributed consensus, IEEE Netw. 33 (2019) 133–139.

[9] H. Jin, Y. Luo, P. Li, J. Mathew, A review of secure and privacy-preserving medical data sharing, IEEE Access 7 (2019) 61656–61669.

[10] M.S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the Internet of Things: A comprehensive survey, IEEE Communications Surveys & Tutorials 21 (2019) 1676–1717.

[11] M.H. Kassab, J. DeFranco, T. Malas, P. Laplante, G. destefanis, V.V. Graciano Neto, Exploring research in blockchain for healthcare and a roadmap for the future, IEEE Trans. Emerg. Top. Comput. (2019) 1.

[12] I. Abu-Elezz, A. Hassan, A. Nazeemudeen, M. Househ, A. Abd-Alrazaq, The benefits and threats of blockchain technology in healthcare: A scoping review, Int. J. Med. Inf. 142 (2020) 104246-104255.

[13] Y. Chen, S. Ding, Z. Xu, H. Zheng, S. Yang, Blockchain-based medical records secure storage and medical service framework, J. Med. Syst. 43 (2018) 5–14.

[14] A.F. Hussein, N. ArunKumar, G. Ramirez-Gonzalez, E. Abdulhay, J.M.R.S. Tavares, V.H.C. de Albuquerque, A medical records managing and securing blockchain based system supported by a genetic algorithm and discrete wavelet transform, Cogn. Sys. Res. 52 (2018) 1–11.

[15] M. Shen, Y. Deng, L. Zhu, X. Du, N. Guizani, Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach, IEEE Netw. 33 (2019) 27–33.

[16] V. Patel, A framework for secure and decentralized sharing of medical imaging data via blockchain consensus, Health Inform. J. 25 (2019) 1398–1411.

[17] S. Cao, X. Zhang, R. Xu, Toward secure storage in cloud-based ehealth systems: A blockchain-assisted approach, IEEE Netw. 34 (2020) 64–70.

[18] S. Wang, D. Zhang, Y. Zhang, Blockchain-based personal health records sharing scheme with data integrity verifiable, IEEE Access 7 (2019) 102887-102901.

[19] D.C. Nguyen, P.N. Pathirana, M. Ding, A. Seneviratne, Blockchain for secure EHRs sharing of mobile cloud based E-health systems, IEEE Access 7 (2019) 66792–66806.

[20] A.A. Omar, M.Z.A. Bhuiyan, A. Basu, S. Kiyomoto, M.S. Rahman, Privacy-friendly platform for healthcare data in cloud based on blockchain environment, Future Gener. Comput. Syst. 95 (2019) 511–521.

[21] R. Akkaoui, X. Hei, W. Cheng, Edgemedichain: A hybrid edge blockchain-based framework for health data exchange, IEEE Access 8 (2020) 113467-113486.

[22] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, N. Yu, Healthchain: A blockchain-based privacy preserving scheme for large-scale health data, IEEE Internet Things J. 6 (2019) 8770–8781.

[23] X. Liu, Z. Wang, C. Jin, F. Li, G. Li, A blockchain-based medical data sharing and protection scheme, IEEE Access 7 (2019) 118943-118953.

[24] A.R. Rajput, Q. DrugLi, M. Taleby Ahvanooey, I. Masood, EACMS: Emergency access control management system for personal health record based on blockchain, IEEE Access 7 (2019) 84304–84317.

[25] F. Jamil, L. Hang, K. Kim, D. Kim, A novel medical blockchain model for drug supply chain integrity management in a smart hospital, Electronics 8 (2019) 505–537.

[26] H. Liu, D. Han, D. Li, Fabric-iot: A blockchain-based access control system in IoT, IEEE Access 8 (2020) 18207–18218.

[27] J. Chi, Y. Li, J. Huang, J. Liu, Y. Jin, C. Chen, T. Qiu, A secure and efficient data sharing scheme based on blockchain in industrial internet of things, J. Netw. Comput. Appl. 167 (2020) 102710-102720.

[28] D. Nuñez, I. Agudo, J. Lopez, Proxy re-encryption: Analysis of constructions and its application to secure access delegation, J. Netw. Comput. Appl. 87 (2017) 193–209.

[29] M. Aminian, H.R. Naji, A hospital healthcare monitoring system using wireless sensor networks, Int. J. Adv. Netw. Appl. 04 (2013) 103–106.

[30] Jara J. Antonio, Zamora-Izquierdo, A. Miguel, Skarmeta, F. Antonio, Interconnection framework for mhealth and remote monitoring based on the internet of things, Sel. Areas Commun. 31 (2013) 47–65.
[31] R.K. Kodali, G. Swamy, B. Lakshmi, An implementation of IoT for healthcare, in: Recent Advances in Intelligent Computational Systems (RAICS), IEEE, India, 2015, pp. 411–416.
[32] Z. Pang, L. Zheng, J. Tian, S. Kao-Walter, E. Dubrova, Q. Chen, Design of a terminal solution for integration of in-home health care devices and services towards the Internet-of-Things, Enterp. Inf. Syst. 9 (2015) 86–116.
[33] M.M. Rathore, A. Ahmad, A. Paul, J. Wan, D. Zhang, Real-time medical emergency response system: Exploiting IoT and big data for public health, J. Med. Syst. 40 (2016) 283–293.
[34] J. Rei, C. Brito, A. Sousa, on Collaboration, Assessment of an IoT platform for data collection and analysis for medical sensors, in: 4th International Conference and Internet Computing (CIC), IEEE, USA, 2018, pp. 405–411.
[35] J. Chen, K. Li, Z. Tang, K. Bilal, K. Li, A parallel patient treatment time prediction algorithm and its applications in hospital queuing-recommendation in a big data environment, IEEE Access 4 (2016) 1767–1783.
[36] J. Chen, K. Li, H. Rong, K. Bilal, K. Li, A disease diagnosis and treatment recommendation system based on big data mining and cloud computing, Inform. Sci. 435 (2018) 124–149.
[37] I.H. Khan, M. Javaid, Big data applications in medical field: A literature review, J. Ind. Integr. Manag.-Innov. Entrepreneurship 06 (2021) 53–69.
[38] P. Yuan, X. Xiong, L. Lei, K. Zheng, Design and implementation on hyperledger-based emission trading system, IEEE Access 7 (2019) 6109–6116.

**Zeng Chen** received the B.S. degree from Chongqing University (CQU) of China in 2018. He is currently pursuing the Master degree at CQU. His research interests include the medical Internet of things (MIoT) and blockchain.

**Weidong Xu** is a graduate student at Chongqing University (CQU) in China. He received the B.S. degree from CQU in 2018. His research interests include text mining and data analysis.

**Bingtao Wang** received the B.S. degree from Chongqing University of China in 2019. He is currently pursuing the Master degree at CQU. His research interest is the application of the Internet of things.

**Dr. Hua Yu** is currently a full professor affiliated with College of Optoelectronic Engineering, Chongqing University. He received his Ph.D. degree from Huazhong University of Science & Technology, China. He is a head of Department of Electronics Science & Technology. He is an IEEE member, senior member of Chinese Institute of Electronics, senior member of Chinese Society of Micro-Nano Technology, Committee member of the Youth Committee of Chinese Society of Micro-Nano Technology, Council member of China instruments and Control Society. His main research interests focus on the IoT, Big data analysis and Sensors.