

Journal Pre-proof

Blockchain-based eHealth system for auditable EHRs manipulation in cloud environments

Haiping Huang, Xiang Sun, Fu Xiao, Peng Zhu, Wenming Wang



PII: S0743-7315(20)30385-3
DOI: <https://doi.org/10.1016/j.jpdc.2020.10.002>
Reference: YJPDC 4308

To appear in: *J. Parallel Distrib. Comput.*

Received date : 30 November 2019
Revised date : 12 August 2020
Accepted date : 5 October 2020

Please cite this article as: H. Huang, X. Sun, F. Xiao et al., Blockchain-based eHealth system for auditable EHRs manipulation in cloud environments, *Journal of Parallel and Distributed Computing* (2020), doi: <https://doi.org/10.1016/j.jpdc.2020.10.002>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier Inc.

Blockchain-Based eHealth System for Auditable EHRs Manipulation in Cloud Environments

Haiping Huang^{1,3}, Xiang Sun^{1,3}, Fu Xiao^{1,3}, Peng Zhu^{1,3}, Wenming Wang^{1,2,3}

¹School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu China

²School of Computer and Information, Anqing Normal University, Anqing 246011, Anhui, China

³Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210023, Jiangsu China

Corresponding Author: Haiping Huang (hhp@njupt.edu.cn)

Abstract—The development of cloud-assisted electronic health system effectively addresses the drawbacks of traditional medical management system. However, some challenging problems such as security and privacy in data storage and sharing cannot be ignored. First, it is difficult to ensure the integrity of electronic health records (EHRs) during the data outsourcing process. Second, it is difficult to guarantee the privacy and traceability of EHRs during the data sharing process. In this paper, a blockchain-based eHealth system called BCES is proposed to ensure that the manipulation of EHRs can be audited. In BCES, each legitimate query manipulation of data consumers, together with each legitimate outsourcing manipulation of hospitals, will be written into the blockchain as a transaction for permanent storage, which ensures the traceability. At the same time, the attributes-based proxy re-encryption is adopted to achieve fine-grained access control of medical data, and any behavior that threatens the integrity of EHRs will be discovered by the auditor. Due to the traceable and tamper-resistant characteristic of blockchain, any entity that had an illegal manipulation of EHRs will be held accountable to the evidence of our constructed Proof-Chain. Finally, security analysis and performance evaluation demonstrate that this scheme is secure and efficient.

Keywords—Blockchain, cloud computing, electronic health records, data outsourcing and sharing

I. INTRODUCTION

Applying Cloud Computing and Internet of Things technologies in medical and diagnostic services industry has already shown great advantages to improving the quality of services. Among the numerous existing schemes, the cloud-assisted electronic health system (eHealth System) has the most prominent manifestation [1], [2]. Compared with the traditional paper-based systems, eHealth systems provide a more flexible, efficient and convenient platform for storing and processing electronic health records (EHRs) from different medical institutions [3]. There is no doubt that the implementation of the cloud-assisted electronic health system will greatly change the current medical applications. Specifically, eHealth System will allow medical institutions to outsource patients' EHRs to cloud servers and establish flexible access control mechanism without incurring significant storage and maintenance costs [4].

Although the superiority of the cloud-assisted electronic health system is obvious, as an emerging application, there will inevitably be some security and privacy issues [5]. The medical institutions, such as hospitals, usually maintain the primary stewardship of eHealth system [6]. Therefore, the pre-processing and outsourcing process of EHRs are usually completed by hospitals once authorized by

patients. Patients can only access their EHRs during the interaction with hospitals. Most of the other time, patients have no control over their EHRs. **Therefore, in order to improve the quality of medical services, allowing users to monitor their own data at any time is an important issue that needs to be resolved in current stage of researches.**

Meanwhile, cloud service providers are unwilling to invest too much money and equipment to protect the privacy of patients' EHRs out of their Service Level Agreements (SLA). They are simply committed to protecting data privacy as much as possible [7]. A single point of failure of the cloud server is also common in reality, and once the failure occurs, the integrity of EHRs is bound to be endangered [8]. **Therefore, excessive trust cannot be placed on cloud service providers, the design of new strategies to break down cloud service providers' right to control EHRs also becomes the focus of this paper.**

A few existing schemes assume that cloud servers will not collude with authorized hospitals to tamper with patients' EHRs. However, this assumption does not conform to the real situation in a sense. As commercial organizations, cloud service providers tend to compromise with interests. Therefore, it is difficult to deal with such trouble without introducing a fully trusted third-party regulator.

Another concern in cloud-assisted electronic health systems is how to provide a controlled, cross-domain and flexible data sharing of EHRs. The dissemination of EHRs has been considered to be a breakthrough for the discovery of new techniques and new therapies for curing diseases [9]. However, on one hand, the current cloud service providers haven't achieved a satisfactory data sharing mechanism; on the other hand, data sharing at the current stage is highly likely to cause the risk of privacy leakage [10]. To address this problem, attribute-based cryptosystem is embedded in cloud data sharing schemes [11], [12], [13], which can realize one-to-many cloud data sharing [14], [15], and any individual who satisfies the attribute condition can access the source data. However, in the cloud-assisted electronic health system, patients, as data owners, usually do not want visitors who meet the attribute conditions to arbitrarily access their valuable EHRs, and they expect each visit must be authorized. Therefore, it is challenging to implement fine-grained access control and ensure that data transactions are legal and traceable.

Blockchain is a decentralized distributed database that creates a completely trusted environment between unfamiliar individuals without third-party trust endorsements [16]. Furthermore, blockchain combined with cryptography technology can ensure transaction traceability, irreparable modification, non-repudiation, support data security sharing and large-scale collaborative computing, as well as privacy protection for users' identity and data [17]. In order to further develop the potential of the blockchain, many emerging technologies have also been applied to blockchain systems, such as edge computing [18]. A blockchain system combining multiple technologies will achieve greater scalability, security and efficiency. Due to the characteristic of the hash function, once the data is written into the blockchain through the consensus mechanism, no one can modify or forge the data. Therefore, the blockchain technology has been seen as a powerful tool to addressing the above problems in cloud-

assisted electronic health systems through its attractive features [19]. However, many current studies on blockchain-based medical systems simply focus on the storage of data and the blockchain is only regarded as a distributed database, which haven't give play to the potential value. Furthermore, incomplete secure data sharing and a large amount of data redundancy makes many schemes unable to be applied in practice. Since current verification schemes usually involve complex calculations, patients with low computing power are always struggling to pay high computational costs to check the accuracy of their data.

Based on the foregoing overview on progresses recorded and challenges in current researches, a blockchain-based eHealth system called BCES is proposed, in order to achieve integrity of EHRs and support the data sharing of confidential EHRs. BCES inherits the advantages of traditional electronic medical systems and overcomes the defects mentioned above. Different from existing blockchain schemes, our scheme abandons the idea of using blockchain to store medical data indexes, thereby ensuring data integrity to a certain extent. The innovation of our solution is that the key information of each outsourcing and query operation is regarded as a transaction, and the transaction bill will eventually be written into the blockchain for permanent storage. Any entity with illegal manipulation will be held responsible based on the evidence in the blockchain.

Specifically, the contributions of this paper are as follows:

1) Aiming at the drawbacks of existing medical systems, we designed a novel and efficient blockchain-based eHealth system-BCES. In this scheme, the characteristics of the blockchain are fully utilized to ensure that all kinds of users' manipulation logs of EHRs are traceable and transparent. Moreover, the structure of the Proof-Chain retains a reasonable channel of accountability for vulnerable patient groups.

2) Flexible access control strategies are designed for the secure sharing of EHRs. The introduction of attributes allows users to customize the authorization group, and the introduction of proxy nodes allows patients to successfully complete data authorization without requiring large computing power.

3) Benefits from the security characteristics of the blockchain, our scheme can resist a variety of attacks, such as substitution attack, migration attack, collusion attack and replay attack, etc. And meanwhile, a performance evaluation is conducted to show the feasibility and efficiency of our scheme.

The rest of this paper is organized as follows. Section II reviews related work. Preliminaries are described in Section III. In Section IV, we make the problem statement and clarify the design goals. The specific construction of our scheme is proposed in Section V. Simulation results and discussions are presented in Section VI. Section VII concludes the whole paper.

II. RELATED WORK

Due to the explosive growth of medical data, the traditional isolated medical data management schemes have shown obvious flaws. In order to overcome the shortcomings and improve the quality

of medical services, some interactive medical systems have been implemented [20], [21]. Most of these solutions focus on how to realize the interaction between medical institutions, but pay little attention to the security and privacy threats of EHRs. As the value of medical data continues to grow, the security and privacy issues towards outsourced EHRs are gradually attracting researcher's concerns [22]. Esposito [23] introduced in detail the drawbacks of using only cloud storage technology to establish a data sharing system in the medical field. They proposed the use of blockchain in medical data sharing to solve data security issues. However, in fact, the article does not propose a specific and practical scheme to these challenges. Xia et al. designed a system for effective management and protection of medical records based on blockchain. They ensure data storage security by verifying identities and encryption keys [24]. However, this system did not consider the risk of data leakage during the data sharing process, which made the system unsustainable for applications in reality. Cao S, and Neri. et al. [25] present a secure cloud-assisted eHealth system to prevent outsourced EHRs from modification by using Ethereum. However, it uses a common symmetric encryption algorithm without considering the difficulties of medical data sharing encountered in computation complexity. Zheng X. *et al.* [26] proposed a medical data sharing scheme that combines blockchain, cloud computing, and machine learning. This scheme can easily realize the sharing of medical data between various institutions. However, it cannot verify the integrity of cloud medical data, and data consumers are unable to know whether they have received the correct medical data. Similarly, Ferdous et al. [27] proposed DRAMS, a blockchain-based distributed monitoring infrastructure for distributed access control systems. DRAMS provides a data security solution, but it does not really solve the problem of effective data sharing. The problem of data security can be solved with blockchain, but how to achieve effective access control in the process of data exchange is also a concern worth studying. In this regard, Li et al. put forward a novel patient-centric framework in a semi-trusted server and designed a set of mechanisms for data access control to stored EHRs. They use ABE (Attribute-Base Encryption) technology to encrypt the EHR file of each patient [28]. However, ABE has many disadvantages. Once the user modifies his access strategy, the system will need additional computational overhead to perform attribute revocation and re-encrypt the data, which is a disaster for the computationally disadvantaged patient group [29] [30]. Xia Q. et al. [31] suggested a blockchain-based electronic medical record sharing scheme. This scheme makes full use of the tamper-resistant characteristics of the blockchain to solve the problem of access control of medical data. However, the patient as the data owner cannot monitor the EHRs in real time, which is not in the patient's actual benefits. Particularly, Yang G. et al. [32] also proposed an architecture that implements blockchain technology to the current EHR system, which is compatible with the existing systems. However, health providers alone take the main responsibility for maintaining the blockchain, including creating, verifying and appending new blocks, which will inevitably threaten the efficiency of the entire system.

By analyzing the existing research schemes, it can be found that the combination of blockchain and medical systems has facilitated the enhancement of service quality. However, what cannot be ignored

is that medical systems based on blockchain and cloud storage are still flawed in terms of fine-grained access control and privacy protection of data sharing. In particular, how to let patients grasp the details of the data sharing process in real time is an urgent problem to be solved. And unsatisfactory efficiency or security have become obstacles to the implementation of these schemes. These issues are exactly the concerns of this paper. In our proposed BCES scheme, a Proof-Chain is used to store users' manipulation logs, and the attribute-based proxy re-encryption component are embedded in BCES to realize secure data sharing which is computationally friendly to patients. It is believed that our scheme will make up for the gaps of above-mentioned researches.

III. PRELIMINARIES

A. Bilinear maps

Let G_1 be an additive cycle group and G_2 be a multiplicative group with the same order p , and g is the generator of G_1 . A bilinear map $e: G_1 \times G_1 \rightarrow G_2$ has the following properties:

- 1) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in \mathbb{Z}_n^*$.
- 2) Nondegeneracy: there exists $P, Q \in G_1$, $e(P, Q) \neq 1$.
- 3) Computability: there exists an efficiently computable algorithm for computing.

Bilinear mapping is an important cryptographic tool. It is used by the cryptographic algorithms such as **KeyGen**, **Sign**, and **Verify** mentioned in the following content.

B. Attribute-Based Proxy Re-encryption

Attribute-based proxy re-encryption (ABPRE) combines the characteristics of both attribute-based encryption (ABE) and proxy re-encryption (PRE) [33], which can effectively ensure data security and implement fine-grained access control and encrypted data sharing in the cloud computing environment. An ABPRE scheme is a 6-tuple, which includes the following algorithms:

Setup(λ, S): This algorithm inputs a parameter λ and system attribute set S , then generates the public parameters pp and the master secret key msk .

KeyGen(pp, msk, S_u): This algorithm inputs pp, msk and a set of attributes S_u , then generates a secret key sk_u associated with S_u .

Encrypt($pp, m, (M, \rho)$): This algorithm inputs pp , a message m and a access structure (M, ρ) , then generates a ciphertext CT .

RKGen($pp, sk_u, (M', \rho')$): This algorithm inputs pp, sk_u and a new access structure (M', ρ') , and generates a re-encryption key $rk_{S \rightarrow (M', \rho')}$.

ReEncrypt($rk_{S \rightarrow (M', \rho')}, CT$): This algorithm inputs a re-encryption key $rk_{S \rightarrow (M', \rho')}$, a ciphertext CT , and then checks whether the attributes set S in $rk_{S \rightarrow (M', \rho')}$ satisfies the access structure (M, ρ) specified for CT ; if $S \models (M, \rho)$, it generates a re-encrypted ciphertext CT' , otherwise, it returns false.

Decrypt($sk_{s'}, CT'$): This algorithm inputs a secret key sk_s , a re-encrypted ciphertext CT' , the algorithm first checks whether the attribute set S' satisfies the access structure (M', ρ') , that is, if $S' \models (M', \rho')$, it returns false, otherwise, returns m .

C. Blockchain

Blockchain is a chained data structure consisting of several data blocks connected by hash function. It implements functions such as data verification, sharing, calculation, and storage through a consensus mechanism. The blockchain provides a distributed trust ledger for each participant, and each node or user maintains and stores the exactly same ledger, which ensures that all users and nodes in the corresponding blockchain are completely consistent. In different application scenarios, blockchain can store and process different data.

A typical blockchain structure is shown in Fig 1.

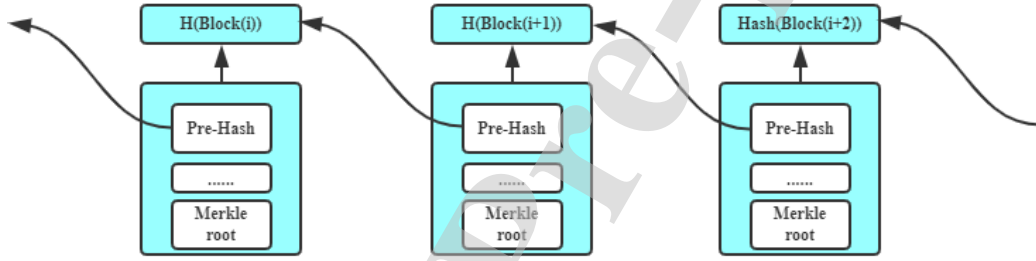


Fig 1. The Structure of Blockchain

D. Attribute Based Signature (ABS)

The blockchain network needs good anonymity. ABS scheme can divide the identity features perfectly, so it can be well applied to distributed networks.

The ABS scheme proposed by Sun Y.*et al.* [34] consists of four algorithms:

Setup(λ, S): This algorithm inputs a parameter λ and generates a system public key pp , a master secret key msk .

KeyGen(pp, msk, S_u): This algorithm inputs pp, msk and a set of attributes S_u , then generates a secret key sk_u , a verification key vk_u associated with S_u .

Sign($pp, m, sk_u, (M, \rho)$): This algorithm inputs pp , a secret key sk_u , a message m , the access structure (M, ρ) , and then returns a signature σ .

Verify($vk_u, S_u, \sigma, (M, \rho)$): This algorithm inputs vk_u , a signature σ , the access structure (M, ρ) and attribute set S_u , and then the signature is valid if **Generate**($vk_u, S_u, \sigma, (M, \rho)$) $\rightarrow \{0,1\}^*$.

IV. PROBLEM STATEMENT

A. Analysis of existing Cloud-Assisted eHealth Systems

The typical system model is shown in Fig 2. Generally, there are five different entities in the model: hospitals, patients, cloud service provider (CSP), research institutions and key server.

As a trusted authority, key server is responsible for distributing key pairs for participating users. CSP stores data from different hospitals and provides an EHRs sharing platform. The research institution sends query requests to the cloud according to its own needs and obtains the query result. Doctors interact with patients and generate EHRs.

Under this system model, there is a hypothesis that patients as data owners lack sufficient **computation**, communication, and storage resources to process their medical data, and the hospital acts as an authorized organization with enough advanced equipment to store and process patients' EHRs. Therefore, patients do not participate in any manipulation of their own medical data during a complete diagnosing, all they need to do is to authorize the hospital to upload EHRs to the cloud. While the patient's EHRs are outsourced to the cloud, the patient also loses control over his or her own EHRs, and the manipulation of the medical data is transparent to the patient.

Though the typical scheme takes advantage of cryptography to defend against external attacks, for some internal attacks such as doctor's tampering with EHRs, there are obvious security flaws.

B. Blockchain-Based eHealth System (BCES)

Designed specifically for user rights, BCES is further optimized on the basis of traditional cloud-assisted electronic health systems, using blockchain to record operational information from data generation to cloud storage.

Our system model consists of seven entities: 1) patients; 2) hospitals; 3) key server; 4) research institutions; 5) cloud service provider (CSP); 6) auditors; 7) Blockchain (as shown in Fig 3). Their respective functions can be described as follows.

- 1) Patients: As the actual owners of EHRs, they are responsible for providing the original medical data and managing their own EHRs remotely.
- 2) Hospitals: Hospitals are delegated by patients to pre-process the EHRs and upload them to the cloud.
- 3) Key Server: Key Servers' duty is to generate and distribute key pairs to other entities.
- 4) Research Institutions: Research institutions are typical data consumers who send query requests to the cloud server and provide their own access structures when needed.
- 5) Cloud Service Provider: The CSP, as a multi-party intermediary, is responsible for storing and transmitting the EHRs.
- 6) Auditor: The auditor is responsible for auditing data security and manipulation security.
- 7) Blockchain: The blockchain is the core of this system model, and the manipulations for each phase of EHRs will be recorded in the blocks. In view of the tamper-resistant and traceable characteristic

of the blockchain, the data stored in the blockchain will be kept as evidence. Once the illegal manipulation is checked in the subsequent verification process, the entity's responsibility can be investigated according to the evidence stored in the blockchain.

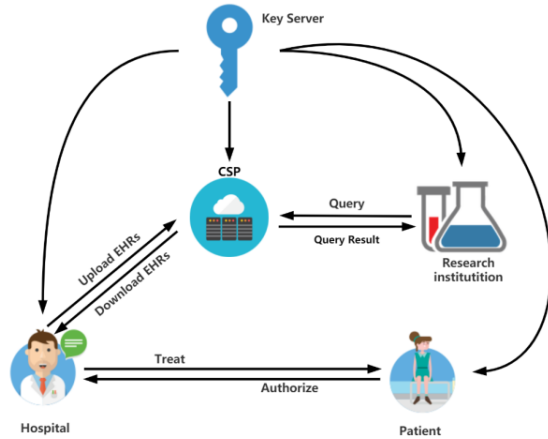


Fig 2. The Typical Cloud-Assisted eHealth System

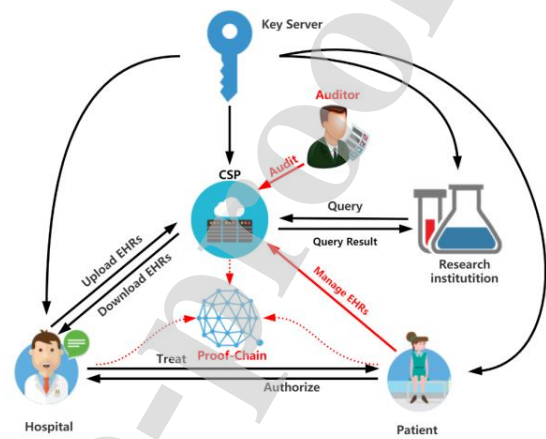


Fig 3. Blockchain-Based eHealth System-BCES

C. Security Model

The key server and auditor in BCES are fully trusted, while the cloud service provider and hospital are considered as semi-trusted. They are **only trustworthy** for a period of time, but in some situations involving their own interests, they may execute some illegal manipulations.

Meanwhile, we assume that our encryption algorithm for EHRs is sufficiently secure. Neither an internal adversary nor an external adversary can crack the ciphertext without obtaining the decryption key, such as modifying the encrypted EHRs. However, the adversaries can make some illegal manipulations on the ciphertext, such as only deleting the ciphertext, issuing query requests without restriction, and substituting the ciphertext with forged one.

D. Design Goals

Under the aforementioned system and **the** security model, our aim is to design a blockchain-based eHealth system for accountability, data owners (patients) are able to detect whether any entity has illegally manipulated their EHRs during each step and to achieve accountability and rights protection. Particularly, the following three properties must be held.

1) Ensure the **security** of the data outsourcing process executed by hospitals. Whether the hospital is trusted or not in the diagnosing period can be judged in our security model. However, the hospital may forge EHRs later to cover their faults made in the diagnosing period due to its semi-trusted characteristic.

2) Ensure the **security** of data storage in the cloud. Most patients' EHRs in the cloud are usually in a state of long-term unused. CSP, for its own benefit, may migrate EHRs that are used at a low

frequency to a server with poor service quality or long service response. This undoubtedly seriously jeopardizes the integrity of EHRs.

3) Ensure the security of the data sharing process to prevent data abuse or theft. EHRs may be used by multiple other unauthorized research institutions under only one valid authorization from data owner. EHRs may also be used by research institutions without authorization.

V. THE PROPOSED SCHEME-BCES

In this section, we introduce the construction and workflow of BCES in detail. Roughly, our scheme can be divided into three parts which include system setting up, data outsourcing and data sharing. To simplify the description, the meaning of some special characters are shown as Table 1.

Table 1
Common notation description

Notations	Descriptions
n	Number of participating users in the system
U_p, U_h, U_r, U_c	Set of different types of users
$addr$	Blockchain account address
g	Generator of cyclic group G_1
k	Number of ciphertext blocks
$O - Tx_s$	Original outsourcing transaction
$F - Tx_s$	Formal outsourcing transaction
$O - Tx_q$	Original query transaction
$O - Tx_q$	Formal query transaction

System Setup

The system setup is performed by the key server which is a fully trusted authority. The key server first needs to initialize the KeyGen algorithm, then lets different types of users join the system, and finally releases private key to users. We give a detailed description of the following two parts:

Setup(λ, S): As shown in Algorithm 1, the key server initializes the setup scheme, generates the public parameters pp and the master secret key msk (line 1). And then the key server initializes the KeyGen scheme, generates a private key sk_{S_i} and a verification key vk_{S_i} for each legitimate participating user and sends it to the user through a secure channel (lines 2-6). It is worth noting that before each generation of the private key, it is necessary to verify whether the user's attribute set meets the requirements.

Algorithm 1 Setup(λ, S)

Input: The security parameter λ ; The attribute set S ;

Output: Deny if the algorithm fails;

1: (pp, msk) \leftarrow **Gen**_{System}(λ, S);

```

2: for  $i = 0; i < n; i++$  do
2:   if  $S_{i \in Z_p} \in S$  then
3:      $(sk_{S_i}, vk_{S_i}) \leftarrow \mathbf{Gen}_{\text{System}}(pp, msk, S_{i \in Z_p})$ ;
4:     send  $(sk_{S_i}, vk_{S_i})$  to the user through a secure channel;
5:   end if
6: end for

```

ClientJoin(λ, ID): Since this system uses the consortium blockchain, a strict admission mechanism must be implemented for each application participant. When the user whose id is ID first joins the system (ID is globally identified), the key server performs a verification operation to determine the validity of the ID information; if the verification result is -1, the application will be discarded (line 1), otherwise the user will be added to user sets according to its key attribute information and meanwhile a blockchain account address $addr$ is created for the user (lines 2-17). The specific steps can be found in Algorithm 2. Suppose there is an applicant trying to join the system, the algorithm first detects whether the applicant's attribute set contains the attribute 'patient' (line 2). If so, the system will create a blockchain account address for the applicant and add the address information to patient set U_p (lines 3-5). The subsequent code has similar functionality as line 2-5 aiming at different entities.

Algorithm 2 ClientJoin(λ, ID)

Input: The security parameter λ ; The user id ID ;

Output: Deny if the algorithm fails;

```

1: if  $\text{Very}(ID) \neq -1$  then
2:   if 'patient'  $\in S_{ID}$  then
3:      $addr_p \leftarrow \text{Hash}(g^{sk_{ID}})$ ;
4:     add  $addr_p$  to  $U_p$ ;
5:   end if
6:   if 'hospital'  $\in S_{ID}$  then
7:      $addr_h \leftarrow \text{Hash}(g^{sk_{ID}})$ ;
8:     add  $addr_h$  to  $U_h$ ;
9:   end if
10:  if 'research'  $\in S_{ID}$  then
11:     $addr_r \leftarrow \text{Hash}(g^{sk_{ID}})$ ;
12:    add  $addr_r$  to  $U_r$ ;
13:  end if
14:  if 'cloud'  $\in S_{ID}$  then
15:     $addr_c \leftarrow \text{Hash}(g^{sk_{ID}})$ ;
16:    add  $addr_c$  to  $U_c$ ;
17:  end if
18: end if

```

Data Outsourcing

In practice, patients usually do not have massive data storage and management capabilities.

Therefore, patients tend to authorize hospitals to complete outsourcing of EHRs.

- After a complete treatment period, patient p generates authorization information as:

$$au_p = \langle Timestamp_p, addr_p, (M_p, \rho_p), FID \rangle$$

Where $Timestamp_p$ denotes the generation time of authorization information, (M, ρ) denotes the access structure embedded in the ciphertext, $addr_p$ denotes the blockchain account address of a specific patient and FID is the unique identifier of the EHR.

- Next, patient p computes $\sigma_{au} \leftarrow \text{Sign}(pp, au_p, (M_p, \rho_p), sk_{s_p, ID})$ and forms the trusted authorization information as ($sk_{s_p, ID}$ denotes the private key of the user who has the '*patient*' attribute and whose identity is ID):

$$au - proof_p = \langle \sigma_{au}, Timestamp_p, addr_p, (M_p, \rho_p), FID \rangle$$

Immediately, patient p transmits the authorization information $au - proof_p$ to the hospital through a secure channel, and at the same time submits it to the blockchain data pool.

- After receiving the authorization information $au - proof_p$ from patient p , hospital h extracts the access structure (M_p, ρ_p) from it and executes Algorithm 3. Due to the consideration for the security of patients' EHRs, EHRs will be divided into several segments based on logical information during the encryption process. As shown in Algorithm 3, the generated ciphertext is a number of ciphertext blocks containing complete logical information as:

$$\{CT_1, CT_2, CT_3, \dots, CT_k\} \leftarrow \text{Encrypt}(m, (M_p, \rho_p), pp)$$

Algorithm 3 EHRs – $\text{Encrypt}(m, (M_p, \rho_p), pp)$

Input: The EHR to be encrypted m ; The Linear integer secret sharing LISS access structure (M_p, ρ_p) ; The public parameters pp ;

Output: Deny if the algorithm fails;

```

1: for  $i = 1; i \leq k; i++$  do
2:    $m_i \leftarrow \text{divideByLogicalInformation}(m)$ ;
3:    $CT_i \leftarrow \text{Encrypt}(m_i, (M_p, \rho_p), pp)$ ;
4: end for
5: return  $\{CT_i\}_{i \in [1, k]}$ 

```

- At the same time, the hospital h generates the original outsourcing transaction bill as shown in Fig 4.

The transaction bill consists of three parts. The first part is the blockchain account information of both clients to this transaction. The second part is charge information, which indicates the cost of this transaction. The third part is data information, which contains the key information *store* –

$proof_h$ submitted by hospital h of this manipulation. It is worth noting that the bill is not the transaction bill which will be eventually written into the block because it lacks the necessary critical information.

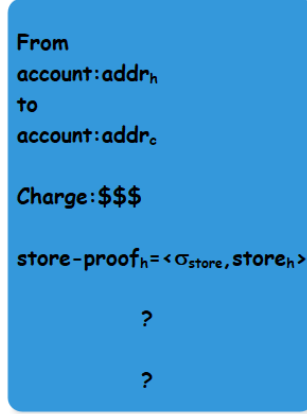


Fig.4. The original outsourcing transaction

The generation process of $store - proof_h$ is shown below.

$$store_h = \langle hash_{CT}, Timestamp_h, addr_p, FID \rangle$$

$$\sigma_{store} \leftarrow \text{Sign}(pp, store_h, (M_h, \rho_h), sk_{s_h, ID})$$

$$store - proof_h = \langle \sigma_{store}, hash_{CT}, Timestamp_h, addr_p, FID \rangle$$

Where $Timestamp_h$ denotes the generation time of outsourcing information, and the calculation process of $hash_{CT}$ is shown in Fig 5.

Subsequently, hospital h will submit the original outsourcing transaction to the blockchain data pool and $\langle \{CT_1, CT_2, CT_3, \dots, CT_k\}, \sigma_{store}, FID \rangle$ will be submitted to the cloud server.

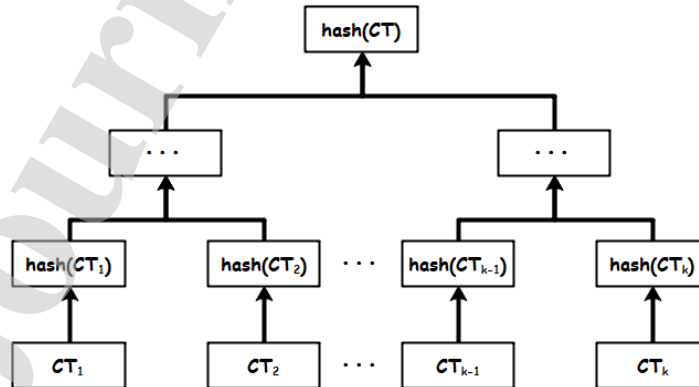


Fig 5. The calculation process of $hash_{CT}$

■ After receiving the EHRs submitted by hospital h , CSP first verifies whether the EHRs are from legitimate users, the specific verification process is shown in Algorithm 4. CSP first verifies whether the attribute information $S_{h,ID}$ conforms to the access structure (M_h, ρ_h) (line 1), and if it does, CSP will generate the signature information and checks whether the correct result is obtained (lines 4-8).

Algorithm 4 σ – $\text{Verify}(vk, S_{h,ID}, (M_h, \rho_h), pp, \sigma_{store})$

Input: The verification key vk ; The attribute set $S_{h,ID}$; The access structure (M, ρ) ; The public parameters pp ; The signature σ_{store}

Output: Deny if the algorithm fails;

```

1: if  $|S_{h,ID}| \neq (M_h, \rho_h)$  then
2:   return null;
3: else
4:   if  $\text{Generate}(vk, S_{h,ID}, (M_h, \rho_h), pp, \sigma_{store}) \rightarrow \{0,1\}^*$  then
5:     return accept;
6:   else
7:     return reject;
8:   end if
9: end if

```

If verification passes, CSP allocates storage space for each ciphertext block as:

$$< mac_1, mac_2, mac_3, \dots, mac_k > \leftarrow \text{Allocate} < CT_1, CT_2, CT_3, \dots, CT_k >$$

Immediately, CSP makes the receiving feedback as:

$$\begin{aligned}
 receive_c &= < hash_{CT}, Timestamp_c, \{mac_i\}_{i \in [1,k]}, addr_c, FID > \\
 \sigma_{receive} &\leftarrow \text{Sign}(pp, receive_c, (M_c, \rho_c), sk_{S_{c,ID}}) \\
 receive - proof_c &= < \sigma_{receive}, receive_c >
 \end{aligned}$$

Then $receive - proof_c$ will be submitted to the blockchain data pool.

■ A formal outsourcing transaction bill will be generated in the blockchain network based on the original outsourcing transaction bill by bookkeeping nodes. The specific generation process of formal outsourcing transaction bill is shown in Algorithm 5. The algorithm first detects the initiator and receiver of the transaction (lines 2-3) and then determines the transaction type. If the initiator is hospital h and the receiver is CSP , the transaction will be determined to be an outsourcing transaction. The bookkeeping node integrates the relevant information from the data pool into a formal outsourcing transaction bill (lines 4-7). A formal outsourcing transaction $F - Tx_s$ will be broadcast to the blockchain network and waiting to be written into the block.

■ The formal outsourcing transaction $F - Tx_s$ will be written into the block via the PBFT consensus mechanism.

Algorithm 5 $F - Tx_s - \text{Gen}(O - Tx_s, au - proof_p, receive - proof_c)$

Input: The original outsourcing transaction bill $O - Tx_s$; The authorization information $au - proof_p$; The receiving feedback $receive - proof_c$;

Output: Deny if the algorithm fails;

```

1: if  $\sigma - \text{Verify}(\sigma_{store}) == \text{accept}$  then
2:   Check the blockchain account of the transaction initiator and receiver;
3:   if  $addr$  of initiator  $\in U_h$  and  $addr$  of receiver  $\in U_c$  then
4:     Extracts  $au - proof_p$  and  $receive - proof_c$  from data pool based on FID;
5:     if  $(\sigma - \text{Verify}(\sigma_{au}) == \text{accept}) \&\& (\sigma - \text{Verify}(\sigma_{receive}) == \text{accept})$  then
6:        $F - Tx_s \leftarrow (\text{add } au - proof_p \text{ and } receive - proof_c \text{ to } O - Tx_s)$ ;
7:       return  $F - Tx_s$  (as shown in Fig. 6);
8:     else
9:       Hold temporarily and process the next transaction;
10:    end if
11:  end if
12: else
13:   Discard this transaction;
14:   return null;
15: end if

```

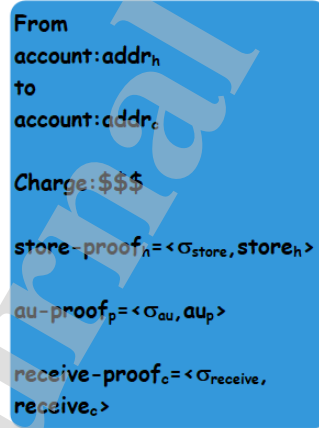


Fig.6. The formal outsourcing transaction

Data Sharing

Medical data is often economically valuable, and any access to medical data must be approved by the data owner. Therefore, in this scheme, a complete data transaction request is as follows:

■ Research institution r initiates data query requests based on their own needs, and initiates a query transaction bill from its client as shown in Fig 7.

The composition of this transaction bill is basically the same as shown in Fig 4, the only difference lies in the part of the data information. It is worth noting that the bill is not the transaction bill which will be eventually written into the block because it lacks the necessary critical information.

The $query - proof_r$ generation process in the transaction bill is as follows:

$$query_r = \langle Timestamp_r, addr_r, addr_p, S_{c,ID}, FID \rangle$$

$$\sigma_{query} \leftarrow \mathbf{Sign}(pp, query_r, (M_r, \rho_r), sk_{S_{r,ID}})$$

$$query - proof_r = \langle \sigma_{query}, query_r \rangle$$

■ This query transaction bill will be submitted to the data pool and then delivered to the client of the specified patient p .

■ If patient p agrees with this query request, he/she will extract the applicant's attribute information $S_{c,ID}$ from the transaction request of the research institution r . Re-encryption key is subsequently generated as:

$$rk_{S \rightarrow (M_r, \rho_r)} \leftarrow \mathbf{RKGen}(pp, sk_{p,ID}, (M_r, \rho_r))$$

The patient will then submit $(rk_{S \rightarrow (M_r, \rho_r)}, S_{p,ID}, FID)$ to the cloud server through a secure channel, and at the same time generate evidence information for this query request, the specific process is as follows:

$$accept_p = \langle Timestamp_p, addr_p, addr_r, FID \rangle$$

$$\sigma_{accept} \leftarrow \mathbf{Sign}(pp, accept_p, (M_p, \rho_p), sk_{S_{p,ID}})$$

$$accept - proof_p = \langle \sigma_{accept}, accept_p \rangle$$

Then $accept - proof_p$ will be submitted to the blockchain data pool.

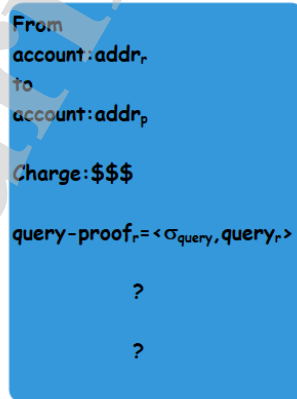


Fig 7. The original query transaction

■ After receiving the re-encryption information from patient p , CSP first checks whether the attributes $S_{p,ID}$ of the patient p meet the access structure embedded in the original ciphertext. If $S_{p,ID} \models (M_p, \rho_p)$ the ciphertext will be re-encrypted as:

$$\{CT_1^*, CT_2^*, CT_3^*, \dots, CT_k^*\} \leftarrow \mathbf{ReEnc}(pp, rk_{S \rightarrow (M_r, \rho_r)}, \langle CT_1, CT_2, CT_3, \dots, CT_k \rangle)$$

otherwise, this query request will be put on hold. And then $\{CT_1^*, CT_2^*, CT_3^*, \dots, CT_k^*\}$ will be sent to research institution r .

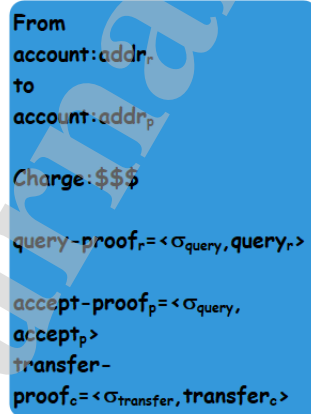
In order to finalize this manipulation, the data transfer certificate needs to be generated as:

$$\begin{aligned} transfer_c &= \langle \text{Timestamp}_c, \text{addr}_c, \text{addr}_r, \text{FID} \rangle \\ \sigma_{transfer} &\leftarrow \mathbf{Sign}(pp, transfer_c, (M_c, \rho_c), sk_{S_{c,ID}})) \\ transfer - proof_c &= \langle \sigma_{transfer}, transfer_c \rangle \end{aligned}$$

Subsequently, $transfer - proof_c$ will be submitted to the blockchain data pool.

■ A formal query transaction bill (in Fig 8) will be generated in the blockchain network based on the original query transaction bill by bookkeeping nodes. The specific generation process of formal query transaction bill is shown in Algorithm 6. The algorithm first detects the initiator and receiver of the transaction (lines 2-3), and then determines the transaction type. If the initiator is research institution r and the receiver is patient p , the transaction will be determined to be a query transaction. And then the bookkeeping node integrates the relevant information from the data pool into a formal query transaction bill (lines 4-7). A formal query transaction $F - Tx_q$ will be broadcast to the blockchain network and waiting to be written into the block.

■ The formal outsourcing transaction $F - Tx_q$ will be written into the block via the PBFT consensus mechanism.



From
account:addr_r
to
account:addr_p
Charge: \$\$\$
query-proof_r = < σ_{query} , query_r >
accept-proof_p = < σ_{query} ,
accept_p >
transfer-
proof_c = < $\sigma_{transfer}$, transfer_c >

Fig.8. The formal query transaction

Algorithm 6 $F - Tx_q \leftarrow \text{Gen}(O - Tx_q, \text{accept} - \text{proof}_p, \text{transfer} - \text{proof}_c)$

Input: The original query transaction bill $O - Tx_q$; The proof of consent $\text{accept} - \text{proof}_p$;
The data transfer certificate $\text{transfer} - \text{proof}_c$;

Output: Deny if the algorithm fails;

```

1: if  $\sigma - \text{Verify}(\sigma_{\text{query}}) == \text{accept}$  then
2:   Check the blockchain account of the transaction initiator and receiver;
3:   if  $\text{addr}$  of initiator  $\in U_r$  and  $\text{addr}$  of receiver  $\in U_p$  then
4:     Extracts  $\text{accept} - \text{proof}_p$  and  $\text{transfer} - \text{proof}_c$  from data pool based on FID;
5:     if  $\sigma - \text{Verify}(\sigma_{\text{accept}}) == \text{accept}$  and  $\sigma - \text{Verify}(\sigma_{\text{transfer}}) == \text{accept}$  then
6:        $F - \text{Tx}_q \leftarrow \text{add } \text{accept} - \text{proof}_p \text{ and } \text{transfer} - \text{proof}_c \text{ to } O - \text{Tx}_q$ ;
7:       return  $F - \text{Tx}_q$  (as shown in Fig 8);
8:   else
9:     Hold temporarily and process the next transaction;
10:  end if
11: end if
12: else
13:   Discard this transaction;
14: return null;
15: end if

```

After a series of outsourcing and query manipulations, we will get the blockchain as shown in Fig 9 and **define it as a Proof-Chain**.

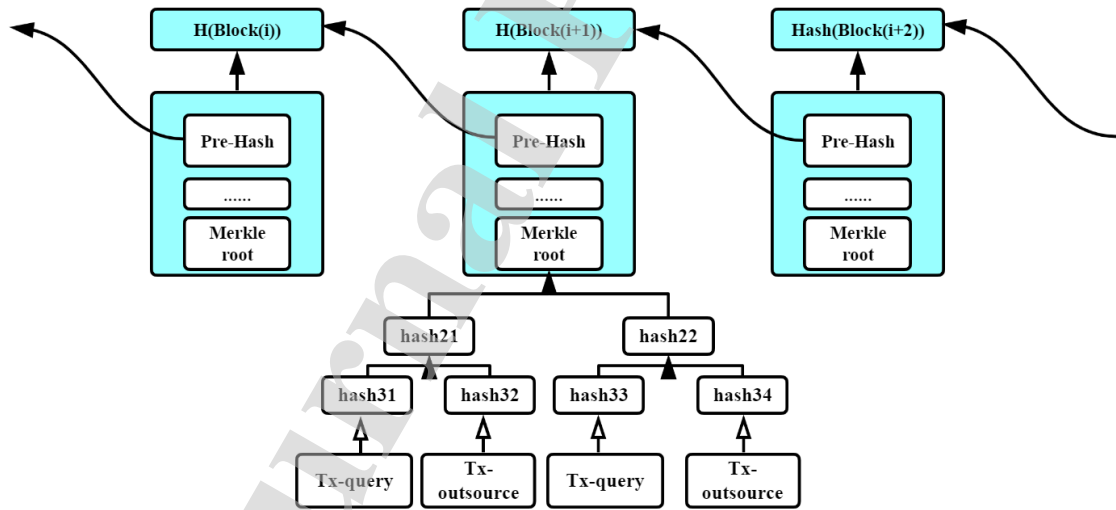


Fig 9. The Structure of Proof-Chain

VI. SECURITY ANALYSIS

We will analyze the security of BCES in terms of the following aspects.

A. Substitution attack

Our scheme can effectively resist substitution attack. In our security model, we assume that hospital

doctors are semi-trusted. In other words, doctors will only remain absolutely honest during the period of interaction with the patient, but if the diagnosing period is completed, the doctor may substitute some EHRs in the cloud to compensate for the faults made in the diagnosing period.

Here we assume that a semi-trusted doctor tries to substitute EHRs blocks containing the diagnostic error with other ciphertext data blocks. If the ciphertext blocks being substituted are (CT_i, CT_j, CT_l) , due to the weak collision of hash function, it is very difficult for this doctor to find three ciphertexts to make the following equations true:

$$\text{Hash}(CT_i) = \text{Hash}(\text{Ciphertext}_1)$$

$$\text{Hash}(CT_j) = \text{Hash}(\text{Ciphertext}_2)$$

$$\text{Hash}(CT_l) = \text{Hash}(\text{Ciphertext}_3)$$

And then auditor a calculates hash'_{CT} according to the calculation process shown in Fig 5. Obviously, due to the weak collision of hash function, we make the inference that $\text{hash}_{CT} = \text{hash}'_{CT}$. Any change in this transaction will inevitably change the Merkle root. Therefore, this situation is never allowed due to the blockchain-based tamper-resist mechanism.

B. Migration attack

Our scheme can effectively resist migration attack. In our security model, CSP is also semi-trusted. On most occasions, CSP will be honest, but sometimes, for its own benefit, CSP may migrate EHRs that are used at a low frequency to a cloud server with poor service quality or long service response.

We assume that the cloud server has migrated some ciphertext blocks (CT_i, CT_j, CT_l) to old and worn-out servers. Once auditor a regularly audits, he will find that the actual server address of (CT_i, CT_j, CT_l) does not match the corresponding transaction content in the blockchain, that is,

$$(\text{mac}_{CT_i}, \text{mac}_{CT_j}, \text{mac}_{CT_l}) = (\text{mac}'_{CT_i}, \text{mac}'_{CT_j}, \text{mac}'_{CT_l})$$

Obviously, this situation is never allowed due to the blockchain-based tamper-resist mechanism.

C. Collusion attack

Our scheme can effectively resist migration attack. There are two types of collusion in this scheme. One is the collusion between hospitals and cloud service provider, and the other one is the collusion between research institutions and cloud service provider. Regarding the first situation, the hash value and location information of EHRs are stored in the blockchain. If malicious behaviors such as deletion, modification, and migration occur, auditor a can detect and hold the corresponding responsibility in time due to the blockchain-based tamper-resist mechanism. Regarding the second situation, because EHRs are encrypted with a specific access policy, data requestors with any attributes that do not meet requirements cannot directly access EHRs. Suppose that research institution r tries to directly obtain EHRs by bypassing the data owner, but the attribute information contained in his private key does not satisfy the access policy embedded in the ciphertext, so he cannot decrypt the ciphertext. At the same time, if the cloud service provider tries to send the re-encrypted ciphertext to an unauthorized entity,

this behavior will not happen, because the cloud service provider cannot obtain the plaintext during the re-encryption process, *CSP* is only responsible for the re-encryption work and re-encrypted EHRs can only be decrypted by authorized entities.

D. Replay attack

Our scheme can effectively resist replay attack. All transactions in the system include the timestamp added by the three entities to this operation, and are accompanied by a digital signature. Since all transactions on the blockchain are transparent, any user can extract the generation time of the transaction. If a malicious user attempts to use the transaction written on the blockchain to repeat the transaction request, then during the transaction verification phase, the relevant verification node can detect that the original transaction time does not match the current time, and will discard the transaction. This mechanism effectively prevents replay attacks.

E. Accountability

Our scheme will provide an approach for accountability. Due to the characteristic of the blockchain, the transaction bills stored in the blockchain are similar to the actual contracts. Once the transaction is successfully written into the blockchain, it means that the two or three entities have reached a consensus on this transaction. If any entity has an illegal operation in the subsequent steps, it will be held accountable.

VII. PERFORMANCE EVALUATION

In this section, we conduct the experiments on an Ubuntu 18.04 with an Intel(R) Core (TM) i5-3230M CPU @2.60GHz 4G memory.

We first test the time cost of the cryptographic algorithms involved in BCES based on cpabe-0.10 library. In this scheme, we choose the prime-order bilinear groups with 128 bits. It is worth noting that this test parameter only serves as a test reference and might not meet the security requirements of the actual system.

Computation overhead

1) Table 2 shows the time cost of several major operations in a 2.6-GHz CPU, 4-GB RAM environment.

We decompose the interaction between users into several key operations as shown in Table 2. A user's interaction consists of the following sub-steps. Among them, G_1 is the multiplication group mentioned in the **PRELIMINARIES** section, which is used in the operations of **KeyGen**, **Sign**, **Verify**, etc.

Table 2.
Time cost of operations

Notations	Definition	Time(ms)
-----------	------------	----------

Mu	Multiplication in G_1	2.4
e	$e: G_1 \times G_1 \rightarrow G_2$	7.9
$hash \rightarrow Z_p^*$	Hash a value into Z_p^*	1.7
Ex	Exponent operation in G_1	1.2
$hash \rightarrow G_1$	Hash a value into G_1	2.0

According to Table 2, we get the relationship between the computation cost and the number of patients during the outsourcing period, as shown in Fig 10.a; and the relationship between the computation cost and the number of research institutions is shown in Fig 10.b.

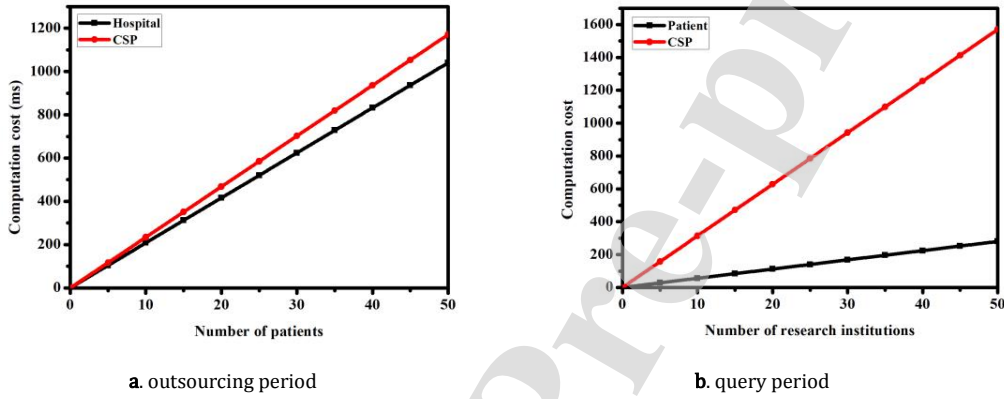


Fig 10. Computation cost during outsourcing period and during query period

As shown in Fig10.a, we tested the computation overhead of hospitals and cloud service providers as the number of patients increased. It is not difficult to find that with the increase in the amount of medical data (the experiment assumes that one patient generates an equal unit amount of medical data), the computational overhead of hospitals and CSPs also show a linear increase. However, it is worth noting that even if the data size (number of patients) reaches 50, the calculation delay of the two is only about 1000ms. For entities equipped with powerful-computation devices, this is a completely acceptable computation delay.

For the situation where multiple research institutions may access the same EHR, relevant experiments are conducted and the obtained data can be shown in Fig 10.b. From Fig 10.b, we can analyze that the computation cost of CSP and patient p increase linearly with the increase in query requests (the experiment assumes that a research institution produces an equivalent unit amount of EHR access). When the access to specific EHRs reaches 50, the computing delay required by CSP is 1500ms. For a cloud server with powerful computing performance, it is a relatively small overhead. At the same time, for a patient with a weak computing power, it takes only 300ms to complete the calculation of the 50 access requests. Obviously, it is feasible for patients to handle concurrent access requests with a simple computation device such as a mobile phone or a personal computer.

In order to show the advantages of BCES in a more intuitive way, the computation power requirements of the entities involved in our scheme are compared with the schemes in [12] and [25], and the comparison results are shown in Table 3 (greater the number of ※, greater the computation power requirement). Since this scheme in [12] lacks a detailed performance evaluation, we can only achieve qualitatively approximate comparisons based on the scheme descriptions.

Table 3.
Comparison in computation power requirement

	Client of patient	Client of hospital	Consensus node
BCES scheme	※	※	※
Scheme in [12]	※	※	※※※
Scheme in [25]	※※	※	※

During the EHRs outsourcing storage period, in our scheme and [12], the patient client only needs to generate authorization information, and does not need to negotiate with the hospital for the treatment key, so the computation power is slightly less than the patient client in [25]. However, the consensus nodes in [12] require higher computation power because they need not only complete the verification of the authorization information signature, but also verify the EHRs themselves, which undoubtedly requires more computation power. And in BCES, consensus nodes only need to verify short information and participate in voting.

Next we will further discuss the quantitative comparisons of the computation cost in the outsourcing process between our scheme and the scheme proposed in [25] under the same experiment conditions. The comparison results are shown in Fig 11.a and Fig 11.b, respectively.

From Fig 11.a and Fig 11.b, it can be clearly seen that during the data outsourcing process, the hospital's computation cost in [25] is slightly less than BCES, but the patient's computation cost is greater than our scheme. The reason for this result is that, in our system, during the interaction with the hospital, the patient does not need to negotiate the treatment key with the hospital. Therefore, a part of the calculation overhead can be reduced for the patient. Obviously, BCES achieves a more satisfactory result from the patient's perspective.

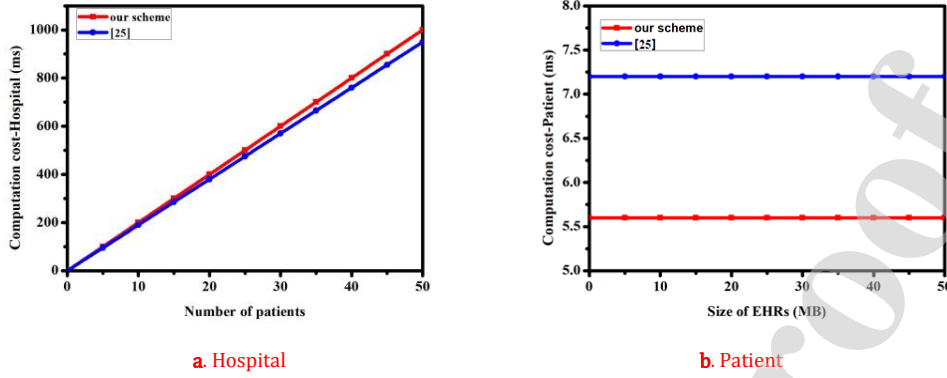


Fig 11. Computation cost comparison

Communication overhead

In this experiment, we tested the communication overhead of patient p and hospital h during the interaction and compared with the scheme proposed in [25].

During the outsourcing phase, for patient p , the communication overhead consists of two parts: one is to delegate hospital h , the other is to submit crucial information to the data pool. For hospital h , the communication overhead also consists of two parts: one is to submit outsourcing transaction bill, the other is to upload EHRs.

In Fig 12, we show the communication cost on patient p and hospital h of two compared schemes, respectively.

From Fig 12, it can be seen that the patient's communication overhead has nothing to do with the size of EHRs, and only needs to pay a small communication cost, where our solution achieves better performance than that in [25] from the patient's perspective. Whether the hospital is in [25] or BCES, the communication overhead increases with the increase in the amount of data, but this is unavoidable because uploading EHRs requires more communication cost. There are tens of thousands of patients compared to the number of hospitals, therefore, reducing communication costs for patients is more valuable than reducing communication costs for hospitals.

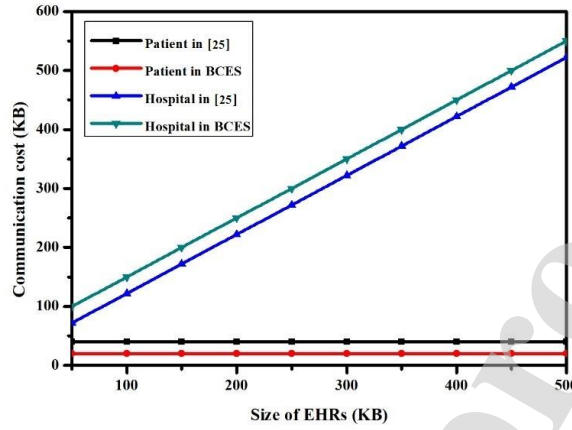


Fig 12. Communication cost on patient p and hospital h .

Consensus overhead

The time overhead of the blockchain system is mainly consumed in the consensus process. In order to further test the time consumption of the entire system, we compared the consensus time cost in our scheme with the consensus algorithms used in [25] and [32], respectively. In the simulation experiments, we set the number of consensus nodes to 20 in the same environment, and run independently using the consensus algorithm in [25], that in [32] and PBFT algorithm in our scheme, and the result after multiple average can be shown in Fig 13.

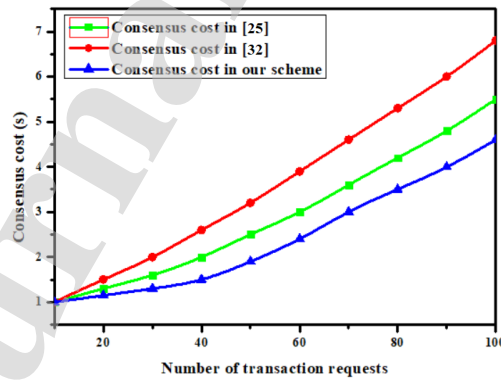


Fig 13. Consensus cost of different schemes

From Fig 13, it can be clearly seen that as the number of transactions increases, the efficiency advantage of the consensus algorithm used in our scheme becomes more obvious. At the same time, the PBFT algorithm avoids the problem of excessive demand for POW computing power, and also

solves the problem of weak centralization of POS, which is more suitable for the application scenarios mentioned in this solution.

VIII. CONCLUSION

The current cloud-assisted electronic medical system combined with the blockchain has some unresolved issues, such as users' inability to actually control their own medical data, hidden safety hazards in EHRs data sharing, and unreasonable protection of the integrity of EHRs. In order to make up for the gaps, a novel blockchain-based eHealth system-BCES was proposed. In this scheme, a Proof-Chain to store various users' manipulations on EHRs was also established. Based on the characteristics of the blockchain, data manipulations can be traced back and cannot be tampered with. The logs stored in Proof-Chain are used as future evidence for rights protection. At the same time, the attribute-based cryptosystem is embedded in the proxy re-encryption mechanism to realize the high security of EHRs sharing, while ensuring that patients with low computing power can also successfully complete EHRs transactions. The security analysis has demonstrated that BCES can defend against different kinds of attacks by fine-grained access control and tamper-resist features. Meanwhile, performance evaluation has been conducted to prove that our scheme is efficient and feasible.

As future work, we hope to improve our design framework through the use of AI and deep learning technology, which will be used to combine different EHRs of the same type to enhance the accuracy of disease diagnosis and treatment without revealing real data information.

ACKNOWLEDGEMENT

This work was supported by the National Key Research and Development Program [grant number 2018YFB0803403]; the National Natural Science Foundation of China [grant number 61672297, and 61872194]; the Key Research and Development Program of Jiangsu Province [grant number BE2017742]; the Postgraduate Research & Practice Innovation Program of Jiangsu Province [grant number KYCX19_0908]; the Key Project on Anhui Provincial Natural Science Study by Colleges and Universities [grant number KJ2019A0579, KJ2019A0554].

REFERENCES

- [1] M. S. Hossain and G. Muhammad, "Cloud-assisted Industrial Internet of Things (IIoT) - Enabled framework for health monitoring," *Comput. Networks*, vol. 101, pp. 192–202, 2016, doi: 10.1016/j.comnet.2016.01.009.
- [2] C. Xu, N. Wang, L. Zhu, K. Sharif, and C. Zhang, "Achieving Searchable and Privacy-Preserving Data Sharing for Cloud-Assisted E-Healthcare System," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8345–8356, 2019, doi: 10.1109/JIOT.2019.2917186.

- [3] V. Casola, A. Castiglione, K. K. R. Choo, and C. Esposito, "Healthcare-Related Data in the Cloud: Challenges and Opportunities," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 10–14, 2016, doi: 10.1109/MCC.2016.139.
- [4] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An Efficient and Secure Deduplication Scheme for Cloud-Assisted eHealth Systems," *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 4101–4112, 2018, doi: 10.1109/TII.2018.2832251.
- [5] J. Xiong *et al.*, "A Personalized Privacy Protection Framework for Mobile Crowdsensing in IIoT," *IEEE Trans. Ind. Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019, doi: 10.1109/tii.2019.2948068.
- [6] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure Attribute-Based Signature Scheme with Multiple Authorities for Blockchain in Electronic Health Records Systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018, doi: 10.1109/ACCESS.2018.2801266.
- [7] F. Armknecht, J. M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2014, pp. 831–843, doi: 10.1145/2660267.2660310.
- [8] S. Fatima and S. Ahmad, "An exhaustive review on security issues in cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 6, pp. 3219–3237, 2019, doi: 10.3837/tis.2019.06.025.
- [9] H. Huang, T. Gong, N. Ye, R. Wang, and Y. Dou, "Private and Secured Medical Data Transmission and Analysis for Wireless Sensing Healthcare System," *IEEE Trans. Ind. Informatics*, vol. 13, no. 3, pp. 1227–1237, 2017, doi: 10.1109/TII.2017.2687618.
- [10] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and Federated Learning for Privacy-preserved Data Sharing in Industrial IoT," *IEEE Trans. Ind. Informatics*, vol. PP, no. 8, pp. 4177–4186, 2019, doi: 10.1109/tii.2019.2942190.
- [11] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," in *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, 2016, pp. 839–858, doi: 10.1109/SP.2016.55.
- [12] H. Wang and Y. Song, "Secure Cloud-Based EHR System Using Attribute-Based Cryptosystem and Blockchain," *J. Med. Syst.*, vol. 42, no. 8, pp. 1–9, 2018, doi: 10.1007/s10916-018-0994-6.
- [13] K. Sethi, A. Pradhan, R. Punith, and P. Bera, "A scalable attribute based encryption for secure data storage and access in cloud," *2019 Int. Conf. Cyber Secur. Prot. Digit. Serv. Cyber Secur. 2019*, pp. 1–8, 2019, doi: 10.1109/CyberSecPODS.2019.8884981.
- [14] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT," *Comput. Networks*, vol. 133, pp. 141–156, 2018, doi: 10.1016/j.comnet.2018.01.036.
- [15] H. Hong, X. Liu, and Z. Sun, "A Fine-Grained Attribute Based Data Retrieval with Proxy Re-Encryption Scheme for Data Outsourcing Systems," *Mob. Networks Appl.*, pp. 1–6, 2018, doi: 10.1007/s11036-018-1102-3.
- [16] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. Ind. Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019, doi: 10.1109/TII.2019.2893433.
- [17] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, 2017, pp. 557–564, doi: 10.1109/BigDataCongress.2017.85.

- [18] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7992–8004, 2019, doi: 10.1109/JIOT.2019.2904303.
- [19] P. T. S. Liu, "Medical record system using blockchain, big data and tokenization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 254–261, doi: 10.1007/978-3-319-50011-9_20.
- [20] W. Wang, H. Huang, Y. Wu, and Q. Huang, "Cryptanalysis and Improvement of an Anonymous Batch Verification Scheme for Mobile Healthcare Crowd Sensing," *IEEE Access*, vol. 7, pp. 165842–165851, 2019, doi: 10.1109/ACCESS.2019.2953042.
- [21] A. H. Krist *et al.*, "Designing a patient-centered personal health record to promote preventive care," *BMC Med. Inform. Decis. Mak.*, vol. 36, pp. 3893–3905, 2011, doi: 10.1186/1472-6947-11-73.
- [22] C. Gao, S. Lv, Y. Wei, Z. Wang, Z. Liu, and X. Cheng, "M-SSE: An effective searchable symmetric encryption with enhanced security for mobile devices," *IEEE Access*, vol. 6, pp. 38860–38869, 2018, doi: 10.1109/ACCESS.2018.2852329.
- [23] R. Surya and G. C. P. Latha, "Blockchain: A panacea for healthcare cloud -based data security and privacy," *Test Eng. Manag.*, vol. 82, pp. 6671–6676, 2020, <https://doi.org/10.1109/MCC.2018.011791712>
- [24] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017, doi: 10.1109/ACCESS.2017.2730843.
- [25] S. Cao, G. Zhang, P. Liu, X. Zhang, and F. Neri, "Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain," *Inf. Sci. (Ny)*, vol. 485, pp. 427–440, 2019, doi: 10.1016/j.ins.2019.02.038.
- [26] X. Zheng, R. R. Mukkamala, R. Vatrappu, and J. Ordieres-Mere, "Blockchain-based personal health data sharing system using cloud storage," 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), 2018, pp. 1–6, doi: 10.1109/HealthCom.2018.8531125.
- [27] M. S. Ferdous, A. Margheri, F. Paci, M. Yang, and V. Sassone, "Decentralised Runtime Monitoring for Access Control Systems in Cloud Federations," *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 2632–2633, 2017, doi: 10.1109/ICDCS.2017.178.
- [28] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, 2013, doi: 10.1109/TPDS.2012.97.
- [29] W. M. Li, X. L. Li, Q. Y. Wen, S. Zhang, and H. Zhang, "Flexible CP-ABE Based Access Control on Encrypted Data for Mobile Users in Hybrid Cloud System," *J. Comput. Sci. Technol.*, vol. 32, no. 5, pp. 974–990, 2017, doi: 10.1007/s11390-017-1776-1.
- [30] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 89–98, 2006, doi: 10.1145/1180405.1180418.
- [31] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Inf.*, vol. 8, no. 2, 2017, doi: 10.3390/info8020044.

- [32] G. Yang and C. Li, "A design of blockchain-based architecture for the security of electronic health record (EHR) systems," *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2018-Decem, pp. 261–265, 2018, doi: 10.1109/CloudCom2018.2018.00058
- [33] S. Luo, J. Hu, and Z. Chen, "Ciphertext policy attribute-based proxy re-encryption," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 401–415, doi: 10.1007/978-3-642-17650-0_28.
- [34] Y. Sun, R. Zhang, X. Wang, K. Gao, and L. Liu, "A decentralizing attribute-based signature for healthcare blockchain," *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, vol. 2018-July, pp. 1–9, 2018, doi: 10.1109/ICCCN.2018.8487349.

A blockchain-based eHealth system BCES is proposed to achieve the traceability and tamper-resist of data manipulation.

An attribute-based proxy re-encryption algorithms (ABPRE) is adopted to achieve fine-grained access control for patients.

Author Biography



Haiping Huang received the B.Eng. degree and M.Eng. degree in Computer Science and Technology from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2002 and 2005, respectively; and the Ph.D. degree in Computer Application Technology from Soochow University, Suzhou, China, in 2009. From May 2013 to November 2013, he was a Visiting Scholar with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K. He is currently a professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include wireless sensor networks and Internet of Things.



Xiang Sun received the B.S. degree from Soochow University, Suzhou, China, in 2017. He is currently pursuing the M.S. degree with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include privacy protection in Internet of things and blockchain.



Fu Xiao received the Ph.D. degree in Computer Science and Technology from Nanjing University of Science and Technology, Nanjing, China, in 2007. He is currently a Professor and Ph.D. supervisor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His main research interest is Wireless Sensor Networks. Dr. Xiao is a member of the IEEE Computer Society and the Association for Computing Machinery.



Peng Zhu received the B.S. degree from Nanjing University of Information Science and Technology, NanJing, China, in 2018. He is currently pursuing the M.S. degree with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include privacy protection in Internet of things and blockchain.



Wenming Wang received the M.S. degree from the College of Information Science and Technology, Jinan University, Guangzhou, China, in 2014. He is currently a Lecturer with the School of Computer and Information, Anqing Normal University. He is pursuing the Ph.D. degree with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China, simultaneously. His research interests include wireless sensor networks and information security.

Author Statement

Haiping Huang: Conceptualization, Methodology, Modelling, Software; Xiang Sun: Data curation, Algorithm design, Writing-Original draft preparation; Fu Xiao: Supervision, Investigation, Validation; Peng Zhu: Software, Visualization; Wenming Wang: Writing-Reviewing and Editing.

Declaration of interests

☒ 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.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

--