Hindawi
Security and Communication Networks
Volume 2021, Article ID 7310771, 10 pages
https://doi.org/10.1155/2021/7310771

*Research Article*

# Efficient and Secure Cross-Domain Sharing of Blockchain Electronic Medical Records Based on Edge Computing

**Yage Cheng**,[1,2] **Bei Gong**,[1,3] **ZhiJuan Jia**,[1,2] **YanYan Yang**,[1,2] **Yuchu He**,[1] **and Xiaofei Zhang**[1]

[1]*College of Information Science and Technology, Zhengzhou Normal University, Zhengzhou 450044, China*
[2]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*College of Computer Sciences, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Bei Gong; gongbei@bjut.edu.cn and ZhiJuan Jia; jzj523@163.com

Received 22 July 2021; Accepted 4 October 2021; Published 19 November 2021

Academic Editor: Xiaolong Xu



In this article, we analysed the problems of electronic medical records (EMRs) and found that the EMRs generated by different hospitals for the same patient are mutually independent and duplication and data sharing are difficult among hospitals. In order to solve this problem, this paper proposes an efficient and secure cross-domain sharing scheme of EMRs based on edge computing. The program allows the doctor to access the personal history EMRs through the patient's authorization so that the doctor can understand the patient's history of illness and, on this basis, generate a new medical record for the patient. Then, the doctor sends the EMRs to the edge server, and the server calculates the ciphertext and adds it to the patient's personal medical record to complete the case update. Analysis shows that this solution can effectively prevent data tampering and forgery through blockchain and avoid privacy leakage problems in plaintext sharing by using searchable encryption and by relying on edge servers to solve nearby computing tasks and divert the computing capacity of cloud servers to improve efficiency. The security proof shows that the scheme satisfies the complex problem of the BDH assumption. Performance analysis shows that the scheme is feasible and efficient.

## 1. Introduction

With the rapid development of the Internet of Things and cloud computing, intelligent systems such as intelligent transportation and smart cities are gradually becoming a hot research topic nowadays [1–3]. At the same time, with the sharp increase in medical demand and the gradual intensification of refined hospital management, the development of the informatization of the medical system is also imperative. Compared with paper medical records, EMRs are related to each other, easy to store, more environmentally friendly, and efficient [4, 5]. It effectively solves the problems of paper medical records [3]. So, it is very popular in hospitals.

However, with the rapid growth of EMRs, the problem of data islands in hospitals has become more prominent. When patients go to different hospitals, each hospital will generate a large amount of EMRs and store them in its own hospital independently, which cannot be shared among them. For doctors, it is impossible to understand the patient's illness history in other hospitals. On this basis, doctors are prone to misdiagnosis and even cause significant problems such as medical malpractice. Moreover, it is also not conducive for the patients to master and understand their health status [5]. In addition, EMRs store the patient's personal privacy information. If they are attacked, they will face security risks, such as privacy leaks [6].

In recent years, blockchain technology has developed rapidly. Due to the characteristics of immutability, data integrity, and distributed storage, blockchain technology has been widely used in all walks of life [7–9]. Since blockchain technology can ensure privacy and security in the

application of EMRs, many scholars have proposed solutions to the current problems of EMRs. Literature [10] proposed blockchain-based healthcare data gateway architecture, enabling the patients to control and share their EMRs easily and securely without violating privacy. It provides a new potential way to improve the intelligence of healthcare systems while keeping patient data private. Literature [11] proposed a blockchain-based EMRs data-sharing framework, using immutability, and built-in autonomy properties of the blockchain sufficiently address the access control challenges associated with sensitive data stored in the cloud. Literature [12] proposed an electronic medical care system based on blockchain, which builds an alliance chain among hospitals. Using the practical Byzantine fault-tolerant algorithm reduces the computational power and ensures the safety and stability of the system, and at the same time, it prevents data tampering and privacy leakage. Literature [13] proposes a framework for sharing medical system data services based on blockchain, which does not rely on a trusted third party and realizes safe storage and privacy protection. Literature [14] used attribute-based encryption and identity-based encryption to ensure data privacy and used blockchain techniques to ensure the integrity and traceability of the EMRs. The most significant advantage of blockchain-based EMRs is that users can securely share the EMRs among hospitals and other institutions. However, most of the existing research only discusses the security search and the data sharing without considering establishing system EMRs for individual patients.

In fact, due to the limited storage space, many medical institutions and enterprises store data on cloud servers. However, with the continuous increase of cloud computing data security issues, it is imperative to upload encrypted data to the cloud server. However, it will face the problem of how to implement ciphertext search when data are shared. In this case, searchable encryption technology came into being [15–18]. It supports ciphertext search while ensuring the security of the data sharing, saving a lot of network and computing costs, and making full use of the enormous computing resources of cloud servers to search for keywords on ciphertexts. Therefore, many electronic medical record sharing schemes use searchable encryption technology to realize ciphertext sharing. Literature [19] proposed a blockchain-based searchable encryption scheme for EMRs. The solution stores the index of EMRs in the blockchain using the blockchain to ensure the integrity, tamper-proof, and traceability of the EMRs index and using searchable encryption to realize ciphertext sharing. Literature [20] constructs a framework based on the blockchain. It uses private chains and alliance chains, combined with searchable technology, to realize the safe search of EMRs while ensuring personal privacy and information security. Literature [21] proposed a blockchain-based secure and privacy-protected EMRs sharing protocol. The scheme mainly uses searchable encryption and proxy reencryption to realize data security, privacy preservation, and access control. Literature [22] combines private chain and consortium chain and uses searchable encryption technology to realize data sharing with significant storage overhead. Literature [23] uses ciphertext strategy attribute-based encryption to encrypt EMRs, and only users with the required attributes can access the data, which can achieve fine-grained access control. The above schemes solved privacy security and ciphertext search through searchable encryption technology but did not consider deduplication.

In response to the above problems, we propose a personal EMRs system with deduplication based on edge server. The plan is to update the EMRs by the doctors in time through the patient's authorization with deduplication and then complete data update. Moreover, it is through blockchain and searchable encryption to ensure data and personal privacy security, and the edge server can offload the computing tasks of cloud services to improve computing efficiency.

## 2. Prerequisite

### 2.1. Bilinearity

*Definition 1.* Suppose $G_1$ is the additive group, $G_2$ is the multiplicative group, and the prime order is $q$. Define a bilinear operation $e$: $G_1 \times G_1 \longrightarrow G_2$ satisfying the following properties [24]:

(1) Bilinear: for any $a, b \in Z_q^*$, there is $e(g^a, g^b) = e(g, g)^{ab}$;

(2) Nondegeneracy: there are $g_1, g_2 \in G_1$ such that $e(g_1, g_2) \neq 1$;

(3) Computable: for any $g_1, g_2 \in G_1$, $e(g_1, g_2)$ can be calculated.

*2.2. Bilinear Diffie–Hellman Hypothesis.* Suppose $G_1$ is the additive group, $G_2$ is the multiplicative group, and the prime order is $q$. Define a bilinear operation $e$: $G_1 \times G_1 \longrightarrow G_2$; $g$ is the generator of group $G_1$. Given a four-tuple $(g, g^a, g^b, g^c)$, it is difficult to calculate $e(g, g)^{abc} \in G_2$.

Suppose algorithm $A$ is used to solve the BDH problem, and its advantage is defined as $\varepsilon$, if $\Pr|A(g, g^a, g^b, g^c) = e(g, g)^{abc}| \geq \varepsilon$.

At present, there is no effective algorithm to solve the BDH problem. Therefore, it can be assumed that the BDH problem is complex [24].

*2.3. Public Key Encryption with Keyword Search (PEKS) Based on Bilinear Mapping.* $H_1$: $\{0,1\}^* \longrightarrow G_1$ and $H_2$: $G_2 \longrightarrow \{0,1\}^{\log p}$ are two hash functions.

(1) KeyGen$(\lambda)$. Randomly select $\alpha \in Z_p^*$ and a generator $g$ of group $G_1$, and output $(sk = \alpha, pk = g^\alpha)$;

(2) Index$(pk, w)$. Randomly select $r \in Z_p^*$ for the keyword $w$, Calculate $t = e(H_1(w), pk^r) \in G_2$ and output index$(pk, w) = (g^r, H_2(t))$;

(3) Trapdoor$(sk, w')$. Using private key $sk$ and keyword $w$ to generate search trapdoor $T_{w'} = H_1(w')^\alpha \in G_1$;

(4) Search$(pk, \text{Index}, T_{w'})$. Set index$(pk, w) = (I_1, I_2)$; check if there is $H_2(e(T_{w'}, I_1)) = I_1$, and output the corresponding index if they are equal [24].

*2.4. System Model.* This paper aims to solve the difficulties in EMRs sharing among hospitals and the problems of isolated and repeated storage of cases. The program mainly uses blockchain and searchable encryption technology to ensure EMRs data and privacy security. The overall idea of the scheme is that when a patient sees a doctor, he first registers with the hospital, and the hospital makes an appointment for the patient. Then, the patient authorizes the doctor to generate EMRs and the doctor sends the EMRs and authorization guarantee to the edge server. The edge server encrypts the EMRs and retrieval information and uploads them to the cloud server and blockchain. When the patient goes to another hospital, the doctor needs to be authorized to visit the personal EMRs. Then, the doctor generates new EMRs after understanding the patient's history of illness and sends them to the edge server. The edge server marks the repeated case and then adds the newly added case to the patient's medical record to complete the case update.

The main entities involved in the system are patients, doctors, hospitals, cloud servers, edge server, and blockchain. The system architecture is shown in Figure 1.

*Definition 2.* The scheme is composed of the following algorithms:

Initialization: generate system parameters;

Key generation: generate the entity's keys;

Registration: the patient registers with the hospital; the hospital makes an appointment for the doctor.

Authorization: the patient authorizes the doctor to generate EMRs.

Generation and storage of electronic medical records: the doctor generates EMRs for the patient and sends them to the edge server. Then, the edge server calculates the ciphertext and index and uploads it;

Access: the doctor views the patient's previous EMRs. The doctor applies for an access request to the edge server and the edge server accesses the blockchain and cloud to obtain the information and then returns it to the doctor.

Update: the doctor deletes duplicate EMRs and sends them to edge server; the edge server updates and uploads them to cloud storage and blockchain.

*2.5. Security Model.* We define the formalized security model of the proposed scheme by the following games.

*2.6. Keyword Privacy Security Game.* If there is no adversary $\mathcal{A}$ who can infer the plaintext of the keywords from the ciphertext or trapdoor in probabilistic polynomial time, the privacy of the keywords can be guaranteed. Define the keywords privacy and security game as follows:

(1) Initialization: given the secure parameter $\lambda$, simulation challenger $\mathcal{B}$ executes the initialization algorithm to generate par.

(2) Phase 1: adversary $\mathcal{A}$ runs the trapdoor generation algorithm multiple times.

(3) Challenge: adversary $\mathcal{A}$ randomly selects two keywords from the keyword space and sends them to the simulation challenger. The simulation challenger executes the trapdoor generation algorithm and then randomly selects a trapdoor and sends it to $\mathcal{A}$.

(4) Guess: After adversary $\mathcal{A}$ inquires $n$ times for the different keywords, it analyzes and guesses. If the $\mathcal{A}$ can guess the trapdoor, then adversary $\mathcal{A}$ wins the game.

*2.7. Proof of Bilinear Diffie–Hellman Hypothesis for Difficult Problems.* If there is an adversary $\mathcal{A}$ who can solve the solution with an advantage $\varepsilon(\lambda)$ in polynomial time, then the adversary $\mathcal{A}$ can solve the BDH difficult problem with an advantage $\varepsilon(\lambda)$ in polynomial time. Define the two-linear Diffie–Hellman hypothesis that the difficult problem specification is proved as follows:

(1) Initialization: given the group $G_1$, $G_2$ and the mapping $e$: $G_1 \times G_1 \longrightarrow G_2$. Simulate challenger $\mathcal{B}$ randomly generates $a, b, c \in Z_p^*$ and sets $g$, $x = g^a$, $y = g^b$, $z = g^c$.

(2) Phase 1: adversary $\mathcal{A}$ runs the encryption algorithm multiple times.

(3) Challenge: the simulate challenger $\mathcal{B}$ randomly selects the plaintext $m$, requires that $m$ is not queried in stage 1, generates the ciphertext $C_m$, and transmits the ciphertext to the adversary $\mathcal{A}$.

(4) Guess: the adversary $\mathcal{A}$ analyzes and decrypts the ciphertext $C_m$. If the adversary $\mathcal{A}$ can decrypt the ciphertext $C_m$ and get the correct plaintext $m$, then the adversary $\mathcal{A}$ wins the game.

(5) Proof: if adversary $\mathcal{A}$ can decrypt the ciphertext, adversary $\mathcal{A}$ can also solve the difficult problem of bilinear Diffie–Hellman assumption.

## 3. The Proposed

The program mainly includes the following essential roles: patients, hospitals, doctors, cloud storage servers, edge server, and alliance chain. The description of symbols in the text is shown in Table 1.

*3.1. Initialization.* The key generation center according to the security parameter $\lambda$ generates the public parameter $\text{par} = \{p, g, G_1, G_2, e, H_1, H_2\}$, where $G_1$ and $G_2$ are the cyclic group of prime order $p$, the generator of group $G_1$ is $g$, $e$ satisfies $G_1 \times G_1 \longrightarrow G_2$, and $H_1$: $\{0,1\}^* \longrightarrow G_1$ and $H_2$: $G_2 \longrightarrow \{0,1\}^{\log p}$ are two hash functions.

*3.2. Key Generation.* The patient $\mathcal{P}$ randomly selects $\alpha \in Z_p^*$ and calculates $h = g^\alpha$, so the keys of $\mathcal{P}$ are $(sk_{\mathcal{P}} = \alpha, pk_{\mathcal{P}} = g^\alpha)$. Similarly, the doctors $\mathcal{D}_1$ and $\mathcal{D}_2$ randomly select $\beta$ and $\gamma$ and calculate $d = g^\beta$ and $f = g^\gamma$, so

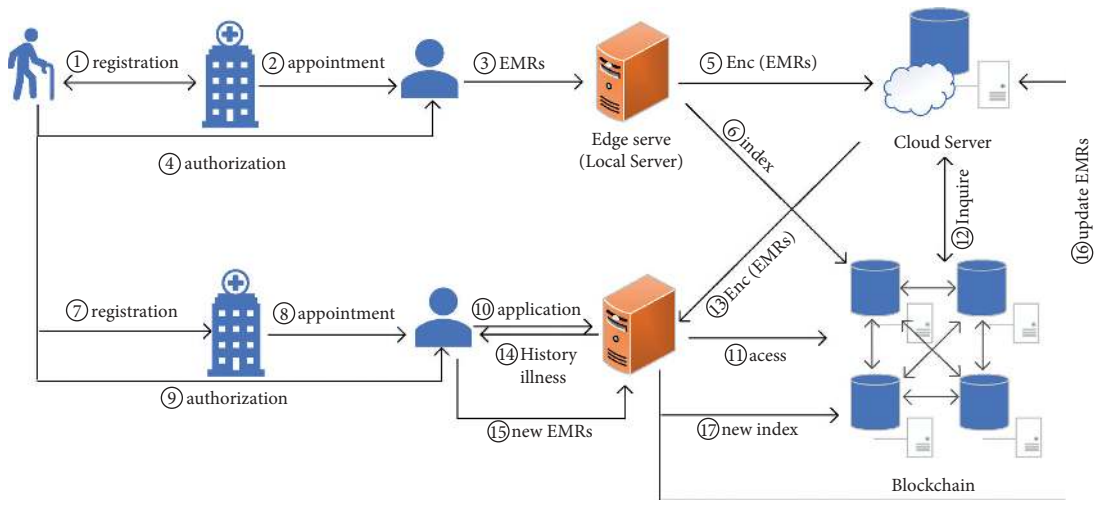


Figure 1: Cross-domain sharing scheme of EMRs.

Table 1: Symbol description.

| Symbols | Roles |
|---|---|
| $\mathscr{P}$ | Patient |
| $\mathscr{H}$ | Hospital |
| $\mathscr{D}$ | Doctor |
| $\mathrm{ID}_{\mathscr{P}}$ | The patient's identification |
| $\mathrm{ID}_{\mathscr{D}}$ | The doctor's identification |
| $\mathscr{CS}$ | Cloud storage server |
| $\mathscr{BC}$ | Alliance blockchain |
| $\mathscr{ES}$ | Edge server (local server of hospital) |
| $\tau$ | Treatment key |
| Aux | Other auxiliary information |
| Dep | Department |
| App | Appointment information |
| Gua | Authorization guarantee |
| $C$ | Ciphertext |
| $T_{\mathrm{Access}}$ | Access time |
| $k$ | Access key |
| $a$ | Repeat mark |
| $t$ | Repeat time |
| $A$ | Access |
| $I$ | Index |
| $T$ | Trapdoor |

the keys of $\mathscr{D}_1$ and $\mathscr{D}_2$ are $(sk_{\mathscr{D}} = \beta, pk_{\mathscr{D}} = g^{\beta})$ and $(sk_{\mathscr{D}_2} = \gamma, pk_{\mathscr{D}_2} = g^{\gamma})$.

*3.3. Registration.* The patient $\mathscr{P}$ registers with the hospital $\mathscr{H}_1$, and the $\mathscr{H}_1$ stores the patient's identification $\mathrm{ID}_{\mathscr{P}}$, randomly selects the treatment key $\tau$, and sends the encrypted $\mathrm{Enc}(pk_{\mathscr{P}}, \tau)$ to $\mathscr{P}$. The hospital $\mathscr{H}_1$ makes an appointment with the attending doctor $\mathscr{D}_1$ for the patient $\mathscr{P}$ and encrypts the appointment information $\mathrm{App} = \mathrm{ID}_{\mathscr{D}_1} \| \mathrm{Dep} \| \mathrm{Aux}$ with the $\tau$ and sends it to $\mathscr{P}$. The patient uses $\tau$ to decrypt the App and obtains the doctor's $\mathrm{ID}_{\mathscr{D}_1}$, department Dep, and other auxiliary information Aux. At the same time, the hospital $\mathscr{H}_1$ sends $\tau$ to the attending doctor $\mathscr{D}_1$.

*3.4. Authorization.* The patient $\mathscr{P}$ authorizes the doctor $\mathscr{D}_1$ to generate EMRs. $\mathscr{P}$ generates an authorization guarantee $\mathrm{Gua}_1 = \mathrm{ID}_{\rho} \| \mathrm{ID}_{D_1} \| T_{\mathrm{Access}_1} \| k_1 \| \tau$, $k_1 \in Z_p^*$, while signing it with the personal private key $\sigma_{\mathrm{Gua}_1} = \mathrm{sig}(sk_{\rho}, \mathrm{Gua}_1)$ and encrypting it with the doctor's public key $C_1 = \mathrm{Enc}(pk_{D_1}, Gua_1 \| \sigma_{Gua_1})$, and then sends $C_1$ to $\mathscr{D}_1$. The doctor $\mathscr{D}_1$ decrypts $C_1$ with the personal private key $sk_{\mathscr{D}_1}$ to obtain the $Gua_1$ and the signature $\sigma_{Gua_1}$, and then the doctor verifies the correctness of the authorization with the patient's public key $pk_{\mathscr{P}}$.

*3.5. Generation and Storage of Electronic Medical Records.* When the verification is passed, the doctor generates EMRs $m_1$ for $\mathscr{P}$ and sends them to edge server. The edge server calculates the ciphertext $C_{m_1} = H_1(k_1)m_1$ and then randomly selects $u \in Z_p^*$ and calculates $I_1 = g^u$, $I_2 = H_2(r)$, $I = \{I_1, I_2\}$, where $r = e(H_1(\mathrm{ID}_{\mathscr{P}}), pk_{\mathscr{D}}^u)$. Finally, it uploads $A_{\mathscr{CS}} = \{C_1, \mathrm{ID}_{\mathscr{D}_1}, I, C_{m_1}\}$ to the cloud server and uploads $A_{\mathscr{BC}} = \{H_1(\mathrm{ID}_{\mathscr{P}}), I_1, I_2, N\}$ to the blockchain, here $N$ is the file number returned by the cloud server.

*3.6. Access.* When $\mathscr{P}$ registers and sees a doctor $\mathscr{D}_2$ in the hospital $\mathscr{H}_2$, $\mathscr{P}$ first authorizes $D_2$ to access his EMRs through the authorization guarantee $Gua_2 = \mathrm{ID}_{\rho} \| \mathrm{ID}_{D_2} \| \tau \| k_1 \| T_{\mathrm{access}}$ and encrypts it as $C_2 = \mathrm{Enc}(pk_{D_2}, Gua_2 \| \sigma_{Gua_2})$, where $\sigma_{Gua_2} = \mathrm{sig}(sk_{\rho}, Gua_2)$. The doctor $D_2$ sends $C_2 = \mathrm{Enc}(pk_{\mathscr{D}_2}, Gua_2 \| \sigma_{Gua_2})$ to edge server, the edge server decrypts $C_2$ with the personal private key $sk_{\mathscr{D}_2}$ to obtain the $Gua_2$ and the signature $\sigma_{Gua_2}$ and then verifies the correctness of the authorization with the patient's public key $pk_{\mathscr{P}}$.

When the verification is passed, the edge server calculates $T = H_1(\mathrm{ID}'_{\mathscr{P}})^{\beta} \in G_1$ and sends $A = \{Gua_2, \mathrm{ID}_{\mathscr{D}_2}, T\}$ to the blockchain nodes. The blockchain nodes execute matching algorithms through $H_2(e(T, I_1)) = I_2$ and return

the corresponding file number $N$. The $\mathscr{CS}$ finds the corresponding ciphertext $C_{m_1}$ through the file number $N$ and returns it to edge server. The edge server sends it to the doctor $\mathscr{D}_2$. $\mathscr{D}_2$ views the patient's history EMRs $C_{m_1}$ by the access key $k_1$ within the limited access time $T_{\text{Access}_1}$.

*3.7. Update.* When the doctor $\mathscr{D}_2$ obtains the patient's EMRs with the access key $k_1$, he first understands the patient's medical history through historical EMRs and generates a new EMRs $m\prime$ on this basis and sends them to edge server. Then, the edge server checks whether the new EMRs have duplicate data by comparing them with the historical EMRs. If there are duplicates, the edge server adds a mark $a$ and a date $t$ based on the historical EMRs and then encrypts the updated EMRs to ciphertext $\text{Enc}(m_2)$ with $k_2$ and adds the newly EMRs to the patient's personal EMRs system in order to complete the update of the EMRs.

When $\mathscr{P}$ registers and sees a doctor $\mathscr{D}_n$ in the hospital $\mathscr{H}_n$, repeat the above process.

## 4. Analysis

*4.1. Correctness*

**Theorem 1.** *In the search phase, the blockchain nodes need to verify the identity of the visitor and secondly verify whether the trapdoor submitted by the edge server has corresponding index and other information, that is, needs to verify whether the equation $H_2(e(T_{\mathscr{CS}}, I_1)) = I_2$ is established. If the equation holds, the corresponding index is returned for the doctor; otherwise, the visit is denied.*

*Proof.* According to the above, we know

$$\begin{aligned} e(T, I_1) &= e\left(H_1(\text{ID}_{\mathscr{P}}')^{\beta}, g^{u}\right), \\ &= e\left(H_1(\text{ID}_{\mathscr{P}}'), g^{u\beta}\right), \\ &= e\left(H_1(\text{ID}_{\mathscr{P}}'), pk_{\mathscr{D}}^{u}\right). \end{aligned} \tag{1}$$

If

$$\text{ID}_{\mathscr{P}}' = \text{ID}_{\mathscr{P}}, \tag{2}$$

then

$$e\left(H_1(\text{ID}_{\mathscr{P}}'), pk_{\mathscr{D}}^{u}\right) = e\left(H_1(\text{ID}_{\mathscr{P}}), pk_{\mathscr{D}}^{u}\right) = r. \tag{3}$$

So,

$$H_2\left(e(T, I_1)\right) = H_2(r) = I_2. \tag{4}$$

Through the proof, we can find that the verification equation is established, the ciphertext retrieval verification is successful, and the result is correct. So, it can retrieve the index information corresponding to the patient's history EMRs, and the correctness of the scheme is verified. □

*4.2. Security.* The scheme satisfies the difficult problem of the BDH assumption; the proof is as follows.

**Theorem 2.** *Assuming that the BDH problem is difficult, the scheme is indistinguishable under adaptive chosen ciphertext attacks (IND-CCA2).*

Suppose $H_1$: $(0,1)^* \longrightarrow G_1$ and $H_2$: $\{0,1\}^* \longrightarrow Z_p^*$ are two random oracles; $\mathscr{A}$ is the adversary of the superior $\varepsilon(k)$ attack scheme. At any time, A can ask $H_1$ or $H_2$ and ask at most $q_{H_1}$ and $q_{H_2}$ times, respectively. Constructing the simulator $\mathscr{B}$ can solve the BDH problem with at least the advantage of $2\varepsilon(k)/eq_{H_2}$ and the running time of $O(\text{time}(A))$.

*Proof.* Suppose the simulator $\mathscr{B}$ has known $g, g^x, g^y, g^z$ $(x, y, z \in Z_p^*)$ and simulate the challenger, with $\mathscr{A}$ as the adversary, and the goal is to calculate $D = e(g,g)^{xyz} \in G_2$.

For simplicity, suppose (1) $\mathscr{A}$ will not initiate the same query to $H_1(\text{ID}_{\mathscr{P}})$ twice, and (2) if $\mathscr{A}$ requests a trapdoor for keyword $\text{ID}_{\mathscr{P}}$, it has already asked $H_1(\text{ID}_{\mathscr{P}})$ before.

(1) System establishment: the simulator $\mathscr{B}$ builds the system, generates the safety parameter $\lambda$, runs the algorithm setup $(1^{\lambda})$, obtains the safety parameter $\text{par} = \{p, g, G_1, G_2, e, H_1, H_2\}$, and generates the keys $K_{\mathscr{C}} = (sk_{\mathscr{C}}, pk_{\mathscr{C}})$ and keeps the private key $sk_{\mathscr{C}}$. The simulator chooses $(x, y, z \in Z_p^*)$, setup $g$, $u_1 = g^x$, $u_2 = g^y$, $u_3 = g^z \in G_1$. The simulator challenger $\mathscr{B}$ returns the parameters Par and the public key $pk_{\mathscr{C}}$ to adversary $\mathscr{A}$, and $\mathscr{A}$ asks the simulator $\mathscr{B}$ with random oracles.

(2) $H_1$ and $H_2$ query: $\mathscr{B}$ randomly chooses $l \in \{1, \cdots, q_{H_1}\}$. $l$ is the guess value of $\mathscr{B}$, and the $l$-th query to $H_1$ corresponds to the final attack result of $\mathscr{A}$. At any time, $\mathscr{A}$ can ask $H_1$ or $H_2$ and ask at most $q_{H_1}$ and $q_{H_2}$ times, respectively.

(1) Inquire $H_1$: $\mathscr{B}$ creates an $H_1^{\text{list}}$, initially empty, and the element is $\langle w_i, h_i, a_i \rangle$. When $\mathscr{A}$ initiates the $i$-th query (set the query value as $w_i$), $\mathscr{B}$ responds as follows:
If $w_i$ is already in the list $H_1^{\text{list}}$, $\mathscr{B}$ takes out the 3-tuple $\langle w_i, h_i, a_i \rangle$ and responds with $H_1(w_i) = h_i \in G_1$. Otherwise, $\mathscr{B}$ chooses a random $a_i \in Z_p$ and calculates as follows: if $i = l$, $\mathscr{B}$ calculates $h_i = y \cdot g^{a_i} \in G_1$; otherwise, $\mathscr{B}$ calculates $h_i = g^{a_i} \in G_1$. Then, $\mathscr{B}$ adds $\langle w_i, h_i, a_i \rangle$ to $H_1^{\text{list}}$ and responds to $\mathscr{A}$ with $h_i$.

(2) Inquire $H_2$: similarly, $\mathscr{B}$ creates a list $H_2^{\text{list}}$ (initially empty) with element type $\langle r_i, v_i \rangle$, $\mathscr{A}$ can query $H_2^{\text{list}}$ at any time, and $\mathscr{B}$ responds as follows:
If $s_i$ is already in $H_2^{\text{list}}$, answer with $H_2(r_i) = v_i$; otherwise, choose $v_i \in \{0,1\}^n$ randomly, answer with $H_2(r_i) = v_i$, and add $\langle r_i, v_i \rangle$ to $H_2^{\text{list}}$.

(3) Trapdoor query (at most $q_{H_1}$ times): when $\mathscr{A}$ requests the trapdoor $T_{\mathscr{CS}}$ corresponding to the keyword $w_i$, let $i$ satisfy $w = w_i$, and $w_i$ represents the query value of the $i$-th query to $H_1$. $\mathscr{B}$ answers the query as follows:

If $i \neq l$, then there is a 3-tuple $\langle w_i, h_i, a_i \rangle$ in $H_1^{\text{list}}$, calculate and return $T_i = u_1^{a_i}$. If $i = l$, then interrupt.

(4) Challenge: $\mathscr{A}$ initiates a challenge. Suppose the keywords of $\mathscr{A}$'s challenge are $w_0$ and $w_1$, and $\mathscr{B}$ randomly selects $J \in \{0,1\}^{\log p}$ and responds with $C = [u_3, J]$.

Note that this response implicitly defines $H_2(e(H_1(w_b), {u_1}^z)) = J$. In other words, $J = H_2(e(H_1(w_b), {u_1}^z)) = H_2(e(yg^{a_b}, g^{ac})) = H_2(e(g^{a_b}, g^{az})^{az(y+a_b)})$. According to this definition, $C$ is a valid trapdoor for the keyword $w_b$.

(5) Trapdoor query: $\mathscr{A}$ can continue to do trapdoor queries for the keyword $w_i$; the only restriction is that $w_i \neq w_0, w_1$, and $\mathscr{B}$ responds as before.

(6) Guess: $\mathscr{A}$ outputs the guess $b' \in (0,1)$, and $\mathscr{B}$ randomly selects $\langle r_i, v_i \rangle$ from $H_2^{\text{list}}$ and outputs $r/e(u_1, u_3)^{a_b}$ as his guess of $e(g,g)^{xyz}$, where $a_b$ is the value used in the challenge phase. This is because $H_2^{\text{list}}$ contains a pair of $\langle r_i, v_i \rangle$, where $r = e(H_1(w_b), u_1^z) = e(g,g)^{az(y+a_b)}$. If $\mathscr{B}$ chooses this pair from $H_2^{\text{list}}$, then $r/e(u_1, u_3)^{a_b} = e(g,g)^{az(y+a_b)}$. The advantage of $\mathscr{B}$ choosing the correct result is $2\varepsilon(\lambda)/eq_{H_2}$, so the probability that $\mathscr{B}$ breaks the security of the proposed scheme is $\Pr|A(r/e(u_1, u_3)^{a_b}) = e(g,g)^{az}(y + a_b)| \geq 2\varepsilon(\lambda)/eq_{H_2}$. □

*4.3. Performance.* By comparing Table 2, we can find that all the above schemes are based on blockchain and realized access control and privacy protection functions. But none of the literatures [11, 20, 22, 23] can implement data deduplication. In addition, reference [11] did not use searchable encryption technology to realize ciphertext search, and reference [20] did not realize data sharing. Therefore, the function of this scheme is better.

Nowadays, there are many researches on EMRs, but it still faces many problems to be solved urgently. For example, we are familiar with privacy protection, access control, and data-sharing issues. With the development of science and technology, more problems have been exposed between the increasing demand of people and the actual status of EMRs. For example, there are no systematic EMRs for patients, and the storage of patients' EMRs is relatively scattered and unsystematic, which makes patients unable to understand personal health systematically. In addition, given the huge data storage and limited storage space of EMRs, deduplication is particularly important. Deduplication can effectively reduce storage consumption and improve storage efficiency. Therefore, it is also one of the urgent problems to be solved in EMRs. In response to the above problems, this article provides some solutions, as shown in the following.

According to Table 3, the plan allows the doctor to update the patient's previous EMRs, so the EMRs system can store the latest medical record in time which ensures the timeliness of the data and realizes integrity and systematic of the patient's EMRs data. Secondly, the deletion of duplicate data effectively improves storage efficiency and reduces storage overhead.

*4.4. Simulation.* The operating system used in the simulation experiment in this article is Windows 10, Intel CPU i7-9750H, and MyEclipse 2015 CI. From the initialization, key generation, encryption, decryption, indexing, and trapdoor generation stages, the execution efficiency of the scheme is investigated. The initialization phase is the configuration of system parameters. The key generation stage is mainly used to generate participants' personal keys. The encryption and decryption use symmetric encryption algorithms. Indexes and trapdoors are used for file query and retrieval. The program selected documents [22, 23] for comparison, and the selected documents were all EMRs sharing schemes based on the blockchain. The comparison results of each stage are shown in Figure 2.

It can be seen from Figure 2 that the execution efficiency of this article is relatively higher than that of documents [22, 23], and documents [22] need to be improved in terms of efficiency. In the index generation stage, the cost of this article is slightly higher than literature [23], while other stages are lower than the comparative literature. This is because literature [23] does not require bilinear operation in the index generation stage, while the solution of this paper needs to perform the bilinear operation, which makes the efficiency relatively lower than literature [23]. In the encryption and decryption stages, literatures [22, 23] require complex operations with the high cost of bilinear pairing and modular idempotence. While this scheme only needs one hash and one inverse operation, computational efficiency is relatively high. In the trapdoor generation stage, the solution in this paper only needs to perform power operation and hash operation, which is more efficient than the comparative literature.

In addition, to further verify the program's performance, the program uses keywords as variables to compare the execution efficiency of the index, trapdoor generation, and search phrases. Figure 3 is the execution time of the index generation phase, Figure 4 is the execution time of the trapdoor generation phase, and Figure 5 is the execution time of the retrieval phase.

It can be seen from Figures 3–5 that with the increase of keywords, the running time of the trapdoor, indexing and retrieval phases in this article, and the comparative literature show an increasing trend. Literature [23] has a higher running time cost with the increase of keywords in the three stages. The running time cost of this article and the literature [22] is relatively consistent, and its execution efficiency is relatively low. Compared with literature [22], the keyword ciphertext matching of this scheme belongs to exact matching, while literature [22] belongs to fuzzy matching. So, the keyword matching result of this scheme is more accurate than literature [22].

Table 2: Function comparison of different schemes.

| Features | Literatures | | | | |
|---|---|---|---|---|---|
| | Literature [11] | Literature [20] | Literature [22] | Literature [23] | This article |
| Blockchain | √ | √ | √ | √ | √ |
| Access control | √ | √ | √ | √ | √ |
| Privacy protection | √ | √ | √ | √ | √ |
| SE | × | √ | √ | √ | √ |
| Data sharing | √ | × | √ | √ | √ |
| Deduplication | × | × | × | × | √ |

Table 3: Problems and the solutions of existing EMRs.

| Types | Problems | Solutions |
|---|---|---|
| Systematisms of personal EMRs | Lack of systematic EMRs for the individuals | By updating EMRs, establishing systematic personal EMRs for patients |
| Privacy leaks | The personal EMRs information of patients is directly shared without encryption or is intercepted or forged by malicious attackers, etc. There is a risk of privacy leakage, and the privacy of patients cannot be guaranteed | Using SE technology to search ciphertexts to avoid privacy leakage, and using blockchain technology to ensure the immutability and integrity of data |
| Store | For the same patient, the same EMRs from the different hospital is repeatedly stored, which makes the storage space consumption high | Delete newly added duplicate data, only mark duplicate EMRs without repetitive storage, and add new cases to the original EMRs |
| Data sharing | The hospitals are relatively independent and have poor interaction. The EMRs of the same patient cannot be shared between hospitals in real time, and there is a problem of data islands | Establish an alliance chain between the hospitals to realize real-time EMRs data sharing |
| Access control permissions | Patients are unaware of personal case sharing and have no access control authority to personal medical records. The hospital can view and share patient data at any time without the patient knowing | Only the doctors authorized by the patient can view and update the patient's personal EMRs |

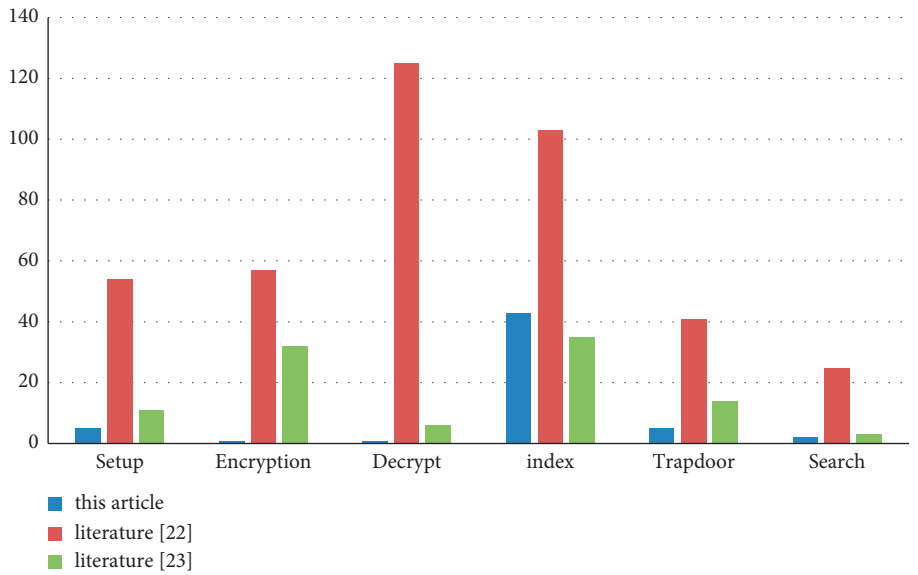


Figure 2: Comparison of the running time of each stage.

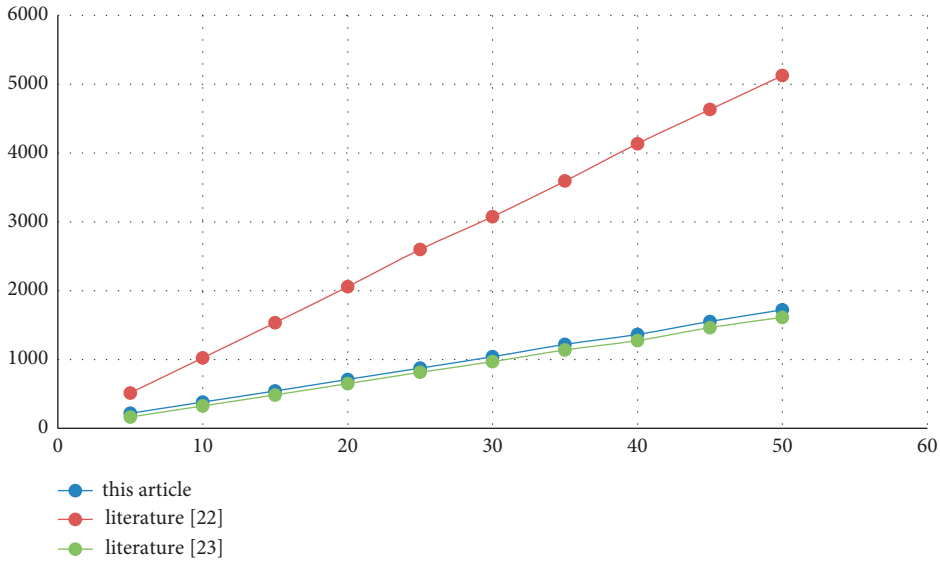


Figure 3: Comparison of index generation time.

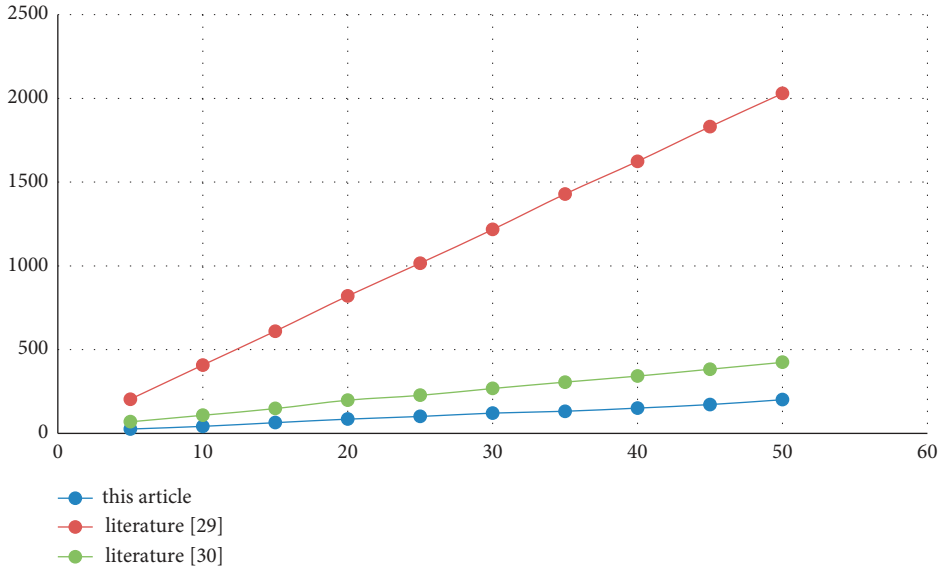


Figure 4: Comparison of trapdoor generation time.

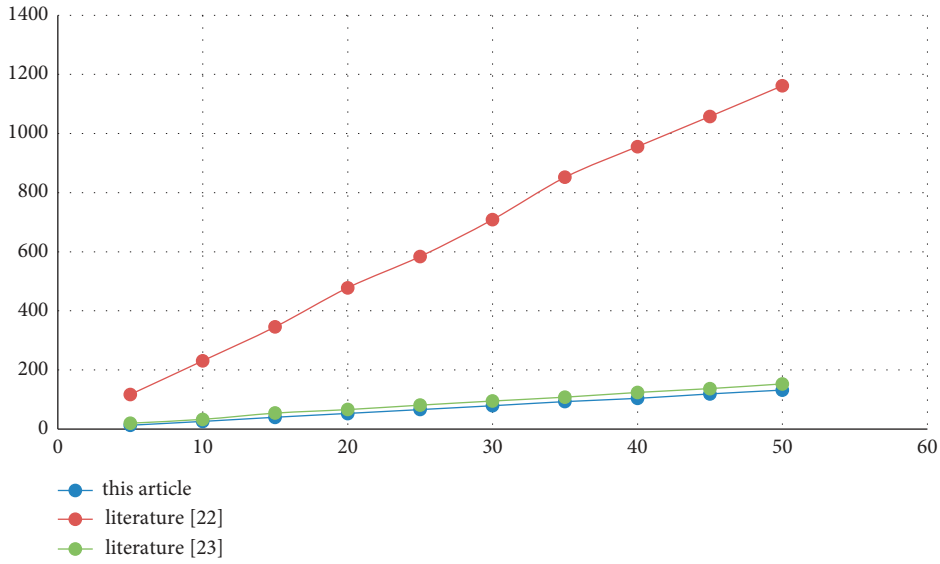


Figure 5: Comparison of retrieval time.

## 5. Conclusions

This article proposes a cross-domain sharing of EMRs among different hospitals based on blockchain and edge computing, which solves the difficulty of EMRs data sharing among hospitals and the problem of isolated and duplicated storage. Through patient authorization, cross-domain secure sharing of EMRs is realized and making the patient's personal EMRs more systematic and complete. The use of blockchain technology ensures that the data cannot be tampered with, and the use of searchable encryption ensures the security of EMRs and personal privacy. Edge servers offload the computing tasks of cloud services and improve computing efficiency. By analysis, it is found that the security of the scheme is proved based on the BDH assumption. Performance analysis and simulation experiments show that the computational complexity is relatively low and has high execution efficiency.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2020-1-09), Henan Key Research Projects of Universities (20A520043 and 21B520022), Natural Science Foundation of Henan Province (202300410510), and National Key Research and Development Program of China (2020YFB1005404).

## References

[1] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.

[2] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[3] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[4] A. Shahnaz, U. Qamar, and A. Khalid, "Using blockchain for electronic health records," *IEEE Access*, vol. 7, Article ID 147795, 2019.

[5] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: towards better research applications and clinical care," *Nature Reviews Genetics*, vol. 13, no. 6, pp. 395–405, 2012.

[6] A. Hoerbst and E. Ammenwerth, "Electronic health records. A systematic review on quality requirements," *Methods of Information in Medicine*, vol. 49, no. 4, pp. 320–36, 2010.

[7] Y. Yuan and F. Y. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481–494, 2016.

[8] Y. Yuan, X. C. Ni, S. Zeng, and F. Y. Wang, "Blockchain consensus algorithms: the state of the artand future trends," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 2011–2022, 2018.

[9] X. Han, Y. Yuan, and F. Y. Wang, "Security problems on blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 45, no. 1, pp. 206–225, 2019.

[10] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of Medical Systems*, vol. 40, no. 10, pp. 218–226, 2016.

[11] Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, "B. B. D. S.: BBDS: blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, pp. 44–60, 2017.

[12] C. Zhang, Q. Li, Z. H. Chen, Z. R. Li, and Z. Zhang, "Medical chain: alliance medical blockchain system," *Acta Automatica Sinica*, vol. 45, no. 8, pp. 1495–1510, 2019.

[13] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *Journal of Medical Systems*, vol. 43, no. 1, pp. 5–14, 2019.

[14] W. Hao and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 18, no. 2, pp. 152–161, 2018.

[15] J. W. Li, C. F. Jia, Z. L. Liu, J. Li, and M. Li, "Survey on the SE," *Journal of Software*, vol. 26, no. 1, pp. 109–128, 2015.

[16] Z. R. Shen, W. Xue, and J. W. Shu, "Survey on the research and development of SE schemes," *Journal of Software*, vol. 25, no. 4, pp. 880–895, 2014.

[17] Y. L. Wang and X. F. Chen, "Research on searchable symmetric encryption," *Journal of Electronics and Information Technology*, vol. 54, no. 10, pp. 2374–2385, 2020.

[18] X. L. Dong, J. Zhou, and Z. F. Cao, "Research advances on secure SE," *Journal of Computer Research and Development*, vol. 54, no. 10, pp. 2107–2120, 2017.

[19] L. Chen, W. K. Lee, C. C. Chang, K. K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Generation Computer Systems*, vol. 95, no. 2, pp. 420–429, 2019.

[20] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, no. 8, pp. 140–158, 2018.

[21] Y. Wang, A. Zhang, P. Zhang, and H. Wang, "Cloud-Assisted EHR sharing with security and privacy preservation via consortium blockchain," *IEEE Access*, vol. 7, no. 2, Article ID 136719, 2019.

[22] S. F. Niu, L. X. Chen, W. T. Li, C. F. Wang, and X. N. Du, "Data sharing scheme based on blockchain," *Acta Automatica Sinica*, vol. 1-11, 2020.

[23] L. Zhang, Z. Y. Zhang, and Y. Yuan, "A controllable sharing model for electronic health records based on blockchain," *Acta Automatica Sinica*, vol. 1-14, 2020.

[24] B. Yang, *Modern Cryptography*, 4th edition, Tsinghua University Press, Beijing, China, 2017.