Blockchain-based Searchable Proxy Re-encryption Scheme for EHR Security Storage and Sharing

Xiaodong Yang*, Xixi Li, Aijia Chen and Wanting Xi

Department of Computer Science and Engineering, Northwest Normal University, room 9c102, Teaching Building No.9, 967 Anning East Road, Anning District, Lanzhou City, China

*Email: y200888@163.com; 1977893583@qq.com

Abstract. The sharing of electronic health records (EHR) has shown significant advantages in the accurate diagnosis of patients and the development of medical institutions. However, due to the privacy and sensitivity of medical data, it is easy to cause security issues such as difficulty in data sharing among different medical institutions and easy leakage of data privacy. Because the blockchain has the characteristics of non-tampering, anonymity, and decentralization. We propose a blockchain-based searchable proxy re-encryption scheme for EHR security storage and sharing. First, we use blockchain and cloud server to store encrypted EHR together to prevent EHR from being tampered with and leaked. Secondly, we use a certificateless encryption and proxy re-encryption based on identity and type scheme as a data sharing protocol. Meanwhile, searchable encryption technology is used to generate a keyword index. Moreover, the proxy node is selected by the delegated proof-of-stake (DPOS) consensus algorithm, which ensures the privacy, immutability and security. It realizes the safe access of third-party data users to medical health data. Finally, security analysis and evaluation show that our scheme can resist identity disguise and replay attacks. In addition, it has stronger security and higher efficiency.

1. Introduction
With the introduction of new reform policy in medical field, the traditional medical care system has been unable to keep up with the pace of contemporary convenient life. The number of electronic health records (EHR) have increased substantially. The emergence of EHR enables patients to have a more comprehensive diagnosis. Moreover, it effectively solves the problems of storage, search, and data sharing of patients’ medical records. It has a broader application value in patient health, hospital development, clinical services and medical field of research [1]. The standardized templates provided by EHR not only can free doctors from the heavy work of writing records, but also assistant doctors to focus on diagnosing and treating of patients [2]. Therefore, carrying out safe storage and effective data sharing, and how to prevent the leakage of data, is always a research hotspot and problem.

Cloud computing as the core business provides an effective method for the sharing of EHR. The hospital outsources the EHR to the cloud server. Only the data users who pass the identity verification can obtain EHR on the cloud server successfully. Biswas et al. [3] believe that storing medical data in a third-party cloud makes it convenient for third-party users to access EHR. However, traditional centralized storage and access control methods are easy to become targets of interest for cyber attackers. Once the cloud server is compromised maliciously, user privacy leaks will be exposed. This is also a very big drawback of the cloud-based electronic medical record data sharing scheme [4].
The decentralization, anonymity, and distributed ledgers of the blockchain provide a platform for EHR to generate permanent and non-tampering records [5]. Han et al. [6] provided practical examples of relevant applications of EHR. Liu et al. [7] proposed a blockchain-based data storage scheme, which stores encrypted EHR in the cloud. It realizes secure sharing and privacy protection. In addition, Yang et al. [8] proposed a multi-keyword scheme based on blockchain. The method ensures the anti-tampering, integrity and traceability. Cui et al. [9] adopts a data sharing scheme that combines blockchain and proxy re-encryption to store encrypted files and related keys in a distributed manner, which can resist collusion attacks between revoked users and distributed proxy. However, storing a number of EHR in a block will reduce the efficiency. So, how to store and share EHR effectively is our facing challenge now.

In order to solve the problem of data sharing and privacy protection of EHR, we propose a blockchain-based searchable proxy re-encryption scheme for EHR storage and sharing, which enables third-party data users to safely access EHR. The contributions of our scheme are as follows.

- We propose the blockchain network and cloud server to store encrypted EHR. The ciphertext is stored on the cloud server. The address of the EHR ciphertext, the keyword index and related medical information are stored on the consortium blockchain. It prevents EHR from being tampered with and leaked, and ensures the anonymity and security.
- We use certificateless public key encryption technology to encrypt data. It can avoid the burden of certificate management and the key escrow problem. Moreover, we use searchable encryption technology to encrypt the keyword index. During search and data sharing, medical data about the data owner is not displayed, ensuring the privacy and security of the data.
- We design a proxy re-encryption technology based on type and identity and use the delegated proof-of-stake (DPOS) consensus algorithm to select proxy nodes, so that no medical institution can infer the user’s true identity from the ciphertext and re-encryption key. It not only reduces the burden of users during data sharing, but also ensures the privacy, reliability and security of data. Our scheme realizes the safe access of third-party data users to EHR.

1.1. Bilinear Maps
Let \( G_i \) and \( G_r \) are two multiplicative cyclic groups with prime order \( p \), where \( e: G_i \times G_i \rightarrow G_r \) is an effective computable map and \( g \) is a generator of \( G_i \). The map is a bilinear map when \( e \) satisfies the following conditions: (1) Bilinear. For any \( a,b \in G_i \), \( i,j \in Z^*_p \), we can get \( e(a^i,b^j) = e(a,b)^{ij} \). (2) Non-degenerate. For \( a,b \in G_i \), and \( e(a,b) \neq 1 \). (3) Completable. For any \( a,b \in G_i \), we can calculate \( e(a,b) \).

1.2. Proxy Re-encryption
Proxy re-encryption is a conversion mechanism used between ciphertexts, which was first proposed by Baize et al. [10] in 1998. The proxy re-encryption solves the inconvenience when users share data. It not only can reduce the burden of users, but also enhance the reliability and security of data. Its participants are: principal, trustee and agent. Each participant cannot obtain any plaintext message, avoiding data leakage and security risks. The trustee delivers its own public key and ciphertext to the network, so that there is no need to repeat the same data sharing multiple times. Our scheme applies the proxy re-encryption to share data, realizing the safe access of third-party data users to EHR.

2. System Model
2.1. System Architecture
Our scheme improves the solution [8]. The system model is shown in Figure 1, it contains six entities, which are KGC, data owner (DO), Medical data provider (MDP), Cloud server (CS), Blockchain and Medical data requester (MDR).
Figure 1. Blockchain-based searchable proxy re-encryption model for EHR storage and sharing.

- **KGC.** KGC is mainly responsible for generating system master keys and public parameters. In addition, KGC generates part of the private key for each data owner, data provider and data user, and sends part of the private key to each entity through a secure channel.

- **Data owner.** The medical data owner represents patients who go to hospitals for treatment, and have their own electronic medical records and data access strategies. First, the data owner should register an account on the consortium blockchain and joins the hospital system. Generally, an authorized doctor generates the EHR for patient and encrypts the original EHR. In addition, the data owner generates a re-encryption key, and distribute to the proxy node.

- **Medical data provider.** The data provider represents the doctor or hospital administrator who manages the EHR. Authorized the MDP encrypted the original EHR and uploaded to the cloud server. Moreover, the data provider stores the keyword index, the account of the DO, and medical information to the blockchain.

- **Cloud server.** The cloud server is honest and semi-trusted. The CS stores the encrypted EHR by the MDP and sends the file address of the EHR to the account of the DO.

- **Medical data requester.** Data requester refer to medical institutions, research institutes, etc. The MDR sends a request to MDP. After obtaining the search trapdoor, authorized MDR can send a verification request to the DO's account address to obtain the re-encrypted EHR. Finally, the MDR uses his private key to decrypt the re-encrypted EHR to obtain the plaintext.

- **Blockchain.** The blockchain assigns an account to DO and MDR. MDP stores the keyword index, the DO's account address and related information to the consortium blockchain. It ensures that the data cannot be tampered with by malicious users. According to the Delegated Proof of Stake (DPOS) consensus algorithm, the miner node that comes out acts as a proxy node to re-encrypt the encrypted EHR.

### 2.2. Protocol Description

#### 2.2.1. System Initialization.** Given a safety parameter k, the system selects \( \hat{e}: G_i \times G_i \rightarrow G_r \), where \( G_i \) and \( G_r \) are multiplicative cyclic groups with order of prime \( p \) and \( g \) is a generator of \( G_i \). Randomly
select \( s \in Z_q^* \) as the master key and calculate \( P = g^s \). Let three anti-collision hash functions \( H_1 : \{0,1\}^* \rightarrow G_1 \), \( H_2 : \{0,1\}^* \times G_1 \rightarrow G_1 \), \( H_3 : \{0,1\}^* \rightarrow Z_q^* \). Then, public the system parameters \( \text{param} = (G_1, G_g, p, g, e, H_1, H_2, H_3) \).

2.2.2. Key Generation. Partial key generation: When receives the DO’s identity \( i \), the MDR’s identity \( j \) and the MDP’s identity \( k \), KGC computes \( psk_i = H_1(id_i) \), \( psk_j = H_1(id_j) \), \( psk_k = H_1(id_k) \). Moreover, KGC sends \( psk_i, psk_j, psk_k \) to DO, MDR and MDP through a secure channel.

Key generation: After receiving partial private key \( psk \), DO randomly selects \( x_i \in Z_q^* \), and calculates \( pk_i = g^{x_i} \). DO sets private key \( sk_i = (x_i, psk_i) = x_iH_1(id_i) \). Similarly, MDR randomly selects \( x_j \in Z_q^* \), and computes \( pk_j = g^{x_j} \) and sets \( sk_j = (x_j, psk_j) = x_jH_1(id_j) \). MDP randomly selects \( x_k \in Z_q^* \). Then, MDP calculates \( pk_k = g^{x_k} \) and sets \( sk_k = (x_k, psk_k) = x_kH_1(id_k) \).

2.2.3. Data Storage. When a patient \( i \) first visits the hospital, patient \( i \) will register an account on the consortium blockchain and join the hospital system. The patient sends the original identity \( i \) to the hospital. Then, the hospital assigns a new identity \( i_{\text{new}} \) and account \( A_i \) to \( i \). Meanwhile, the patient \( i \) sends data packet \( \eta_i = (i_{\text{new}} \| A_i) \) to generates EHR by the authorized doctor \( k \). The doctor \( k \) categorizes the EHR and extracts a series of keywords. Then, the doctor \( k \) encrypts the plaintext \( m \in \{0,1\}^* \) and identification \( id \) with \( pk_i \) and \( pk_k \), and generates the ciphertext \( C_m = (c_1, c_2, c_3, c_4, c_5) \). The ciphertext \( C_m \) is uploaded to CS by doctor \( k \). The detailed encryption process is as follows.

- Randomly choose \( r \in Z_q^* \).
- Compute \( c_1 = g^r \), \( c_2 = m \cdot (P, pk_i)^{-1} \cdot H_1(id_i) \), \( c_3 = rsk_i, c_k = g^{rsk_i} \), \( c_5 = g^s \), where keywords \( W = (w_1, w_2, \ldots, w_n) \) and type \( T \in \{0,1\}^* \).
- Set the ciphertext \( C_m = (c_1, c_2, c_3, c_4, c_5) \).

2.2.4. Keyword Index Generation and Search. Phase 1: Keyword index generation. According to the patient’s EHR, the doctor \( k \) generates keywords \( W = (w_1, w_2, \ldots, w_n) \) and calculates keyword ciphertext \( C_w = (U, V, R, R_1, \ldots, R_n) \). The doctor \( k \) sends the ciphertext data packet \( \eta_1 = (A \parallel C_m \parallel C_w) \) to CS. Moreover, CS sends the file address \( F_i \) to the patient’s account \( A_i \). Meanwhile, doctor \( k \) sends the keyword data packet \( \eta_2 = (A_i \parallel C_i) \) to the consortium blockchain, where \( C_i \) is the signature of the data packet \( \eta_2 \) by doctor \( k \). The description of keyword ciphertext generation is as follows.

- Randomly choose \( \alpha, \beta \in Z_q^* \).
- Compute \( U = g^\alpha \), \( V = pk_i^\beta \), \( R_1 = t_i^{\alpha \beta} + H_1(id_i)^{\alpha \beta} \), \( l_i = H_1(w_i) \), \( h_i = H_2(w_i \| U \| V) \), keywords \( W = (w_1, w_2, \ldots, w_n) \) and \( 1 \leq i \leq n \).
- Set the keyword ciphertext \( C_w = (U, V, R_1, R_2, \ldots, R_n) \).
Phase 2: Keyword search. When MDR \( j \) obtains the patient’s EHR, he first sends a search request to doctor \( k \). Then, the doctor \( k \) generates keywords \( \varphi = (\varphi_1, \varphi_2, \cdots, \varphi_t) \) and a trapdoor function \( T_{\rho} \) for \( j \) with the keyword \( w \) and \( sk_k \). The MDR \( j \) receives \( T_{\rho} \) and searches the specified keywords on the consortium blockchain. The consortium blockchain checks whether the equation (1) holds:

\[
\hat{e}(T_{\rho_i}, \sum_{i=1}^{t} C_i) = \hat{e}(U, T_{\rho_i}) \cdot \hat{e}(V, T_{\rho_i})
\]

where \( U = g^\varphi \), \( V = p^k \). If the equation holds, the patient \( i \) sends account \( A_i \) to \( j \). Otherwise, the request fails. The detailed generation of trapdoor function is as follows.

- Randomly choose \( \lambda \in Z_q^\ast \).
- Compute trapdoor function \( T_{\rho_1} = g^\lambda, T_{\rho_2} = \sum_{i=1}^{t} l_i^\lambda \), \( T_{\rho_3} = \sum_{i=1}^{t} h_i^\lambda (H_i(id_i)^{t-1}) \), where keywords \( \varphi = (\varphi_1, \varphi_2, \cdots, \varphi_t) \), \( l_i = H_i(\varphi_i) \) and \( h_i = H_j(\varphi_i || U || V) \).
- Set trapdoor function \( T_{\rho} = (T_{\rho_1}, T_{\rho_2}, T_{\rho_3}) \).

2.2.5. Data Sharing. When the MDR \( j \) receives the patient’s account \( A_i \), he sends the data packet \( \eta_i = (j \| A_i \| \| pk_j \| \| pk_k) \) to the patient’s account \( A_i \). After receiving the access notification, patient \( i \) sends account packet \( \eta_i = (H_3(w_i) \| F_i) \) to the MDR’s address \( A_j \), and calculates the re-encryption key \( r_{i,j} \) with \( pk_i \), \( pk_j \) and \( pk_k \). The detailed calculation of re-encryption key is as follows.

- Choose keyword hash \( H_5(\varphi_i) \), file address \( F_i \), identification \( id \) and type \( T \in \{0,1\}^* \).
- Compute \( r_k = \frac{1}{sk_k} (\sum_{i=1}^{t} H_3(\varphi_i)) pk_k \), \( r_k = \frac{1}{sk_k} pk_j \cdot H_3(F_i) \cdot H_2(H_i(id_i)||T) H_i(id_i)^t \), where keywords \( \varphi = (\varphi_1, \varphi_2, \cdots, \varphi_t) \) and type \( T \in \{0,1\}^* \).
- Set the re-encryption key \( r_{i,j} = (r_k, r_{k_2} \cdots) \).

The patient \( i \) sends the re-encryption key \( r_{i,j} \) to the proxy to check whether the equation (2) holds:

\[
\hat{e}(c_i, g) = \hat{e}(c_s, r_{k_i})
\]

where \( c_i = g^\lambda \sum_{i=1}^{t} H_1(w_i) \), \( c_s = rsg^x H_2(id_i)^t \). If the equation holds, the proxy generates a re-encrypted ciphertext \( C_m \). Otherwise, the request fails and the proxy is not allowed to re-encrypt \( C_m \). The generation of re-encrypted ciphertext is as follows.

- Choose the ciphertext and the re-encryption key.
- Calculate \( c'_i = c_i = g^\lambda \sum_{i=1}^{t} H_1(w_i) \), \( c_z = c_z = \sqrt[m]{H_2(H_i(id_i)||T)} = \hat{e}(g^\lambda, g^x)^{H_2(H_i(id_i)||T)} \), \( c'_z = c'_z = \hat{e}(g, g)^{\lambda k_h(x, H_i(F_i), H_i(id_i)||T)} \), \( c_z = c_z = (H_3(m)P + P)^t \), \( c_z = c_z = g^\lambda \), where keywords \( w = (w_1, w_2, \cdots, w_t) \), the file address \( F_i \) and type \( T \in \{0,1\}^* \).
- Set the re-encrypted ciphertext \( C_m = (c_1, c_z, c_z, c'_z, c_z) \).
2.2.6. Data Recovery. The proxy node sends the re-encrypted ciphertext $C_m'$ to the account $A_j$. After receiving $C_m'$, the MDR $j$ checks whether the following two equations (3) and (4) hold:

$$
\hat{e}(c_j, g) = \hat{e}(c_j, H_3(\lambda)P + P)
$$

(3)

$$
\hat{e}(c_j, g) = \hat{e}(c_j, \sum_{i=1}^{n} H_3(w_i) \cdot g)
$$

(4)

where $\lambda \in Z_q^*$ and keywords $W = (w_1, w_2, \cdots, w_n)$. If two equation hold, MDR $j$ decrypts the re-encrypted ciphertext $C_m'$ with private key $sk_j$, thereby recovering a valid plaintext $m$.

3. Security Analysis

3.1. Correctness Analysis

The equation (5) for correctness verification of the re-encrypted ciphertext $C_m'$ is shown as follows.

$$
\frac{c_j}{(c_j, H_3(F_j)} = \frac{\hat{e}(g, g)^{rs_{x_j}H_2(H_i(id_i)[F_j]} \cdot \hat{e}(g, g)^{rs_{x_j}H_2(H_i(id_i)[F_j]} \cdot \hat{e}(g, g)^{rs_{x_j}H_2(H_i(id_i)[F_j]} = m
$$

Consequently, the equation (5) shows that the MDR can decrypt the re-encrypted ciphertext with his private key to obtain the plaintext $m$.

3.2. Security Analysis

- **Immutability and integrity.** First, the protocol uses certificateless public key encryption, which effectively avoids the key escrow and the burden of certificate management. The EHR on the blockchain is a data structure based on a hash algorithm and any modification will result in a change in the hash value of the entire block. Moreover, the blockchain has the characteristics of a distributed database, and prevents data loss and ensure data integrity and non-tampering.

- **Privacy protection.** The blockchain enables EHR to be transmitted in the form of ciphertext. Both the patient and data provider send and receive data packets through the account address on the blockchain. User cannot obtain the true identity of each other, which protects data privacy and the true identity of the entity. Meanwhile, any medical institutions cannot infer the patient’s true identity from the ciphertext. It ensures data privacy, reliability and security.

- **Fine-grained access control.** Our protocol uses searchable encryption technology to encrypt the keyword index. MDR sends a search request to the data provider, and the public and private keys of the data provider are included in the keyword ciphertext and trapdoor function. Therefore, the data access is controlled by the data provider. Moreover, the generation of the re-encryption key includes the private key of the patient’s private key, the data user’s public key and the data provider’s public key. Only authorized data users can decrypt the re-encrypted ciphertext. The sharing of data is controlled by the DO, ensuring the controllability.

- **Collusion resistance.** A data user who wants to obtain the EHR plaintext must meet the access control policy and successfully decrypt the data. Assuming the data owner is credible and his identity credentials is never exposed to others. Simultaneously, the data owner’s private key cannot be obtained by the proxy. The proxy cannot decrypt the first-layer ciphertext and re-encrypted ciphertext. Therefore, the decryption is failure and the EHR plaintext cannot be obtained. Therefore, our scheme can effectively resist identity disguise and replay attacks.

4. Performance Evaluation
Our scheme is compared with the literature [9]. We use the symbol $m$ to indicates the length of the encryption key and $n$ is the length of the search keyword. $T_{exp}$, $T_m$, $T_e$ and $T_h$ are used to denote the time required for exponential operation, multiplication operation, hash operation and bilinear pairing operation, respectively. The calculation cost analysis results of each stage are shown in Table 1. Our scheme has higher efficiency in the keyword index generation stage and trapdoor stage.

| Storage                  | [9]            | Our scheme          |
|--------------------------|----------------|---------------------|
| Keygen                   | $4T_{exp} + 2T_h$ | $6T_{exp} + 3T_h$   |
| Index gen                | $(2m + 3)T_{exp} + mT_m + mT_h$ | $4T_{exp} + T_m + 3T_h$ |
| Trapdoor                 | $(2n + 3)T_{exp} + (n + 1)T_m + nT_h$ | $3T_{exp} + 2T_h$ |
| Keyword search           | $T_m + 3T_e$    | $T_m + 3T_e$        |

5. Conclusion
This article solves the problems of easy leakage of medical data privacy and difficulty in data sharing between different medical institutions. Our scheme has all advantages of certificateless encryption, cloud storage and blockchain, which achieves privacy protection, access control, non-tampering and collusion resistance. We adopt proxy re-encryption based on identity and type to ensure security and realize the fine-grained access control. Meanwhile, we ensure the safe access of third-party data users to EHR. Performance evaluation shows that the scheme has higher efficiency. In the future work, we will conduct more comprehensive research from storage overhead and improving consensus efficiency.

References
[1] Yang X D, Li T, Xi W T, Chen A J and Wang C F 2020 A blockchain-assisted verifiable outsourced attribute-based signcryption scheme for EHRs sharing in the cloud J. IEEE Access. 8 170713-31.
[2] Ivan D 2016 Moving toward a blockchain-based method for the secure storage of patient records. Use of Blockchain for Healthcare and Research Workshop (Gaithersburg, Maryland, United States: ONC/NIST) pp 1-11
[3] Liu Y, Zhang Y, Ling J and Liu Z 2018 Secure and fine-grained access control on e-healthcare records in mobile cloud computing J. Future Generation Computer Systems. 78 1020-26
[4] Biswas S, Akhter T, Kaiser M S and Mamun S A 2014 Cloud based healthcare application architecture and electronic medical record mining: An integrated approach to improve healthcare system. In 2014 17th Int. Conf. on Computer and Information Technology (ICCIT). IEEE. pp 286-291
[5] Wang Y, Zhang A, Zhang P and Wang H 2019 Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain J. IEEE Access. 7 136704-19
[6] Han X, Yuan Y and Wang F Y 2019 Security problems on blockchain: The state of the art and future trends J. Acta Automatica Sinica. 1 206-225
[7] Liu J, Li X, Ye L, Zhang H, Du X and Guizani M 2018 BPDS: A blockchain based privacy-preserving data sharing for electronic medical records. In 2018 IEEE Global Communications Conf. (GLOBECOM). IEEE. pp 1-6
[8] Yang X D, Chen G L, Wang M D, Li T and Wang C F 2020 Multi-Keyword certificateless searchable public key authenticated encryption scheme based on blockchain J. IEEE Access. 8 158765-77
[9] Cui S, Asghar M R and Russello G. 2018 Towards blockchain-based scalable and trustworthy file sharing. In 2018 27th Int. Conf. on Computer Communication and Networks (ICCCN). IEEE. 2018 pp 1-2

[10] Blaze M, Bleumer G and Strauss M 1998 Divertible protocols and atomic proxy cryptography. Int. Conf. on the Theory and Applications of Cryptographic Techniques (Berlin, Heidelberg: Springer) 1998 127-144