Hindawi
Security and Communication Networks
Volume 2022, Article ID 6737942, 13 pages
https://doi.org/10.1155/2022/6737942

Research Article

A Blockchain-Assisted Electronic Medical Records by Using Proxy Reencryption and Multisignature

Xiaoguang Liu,[1,2] **Jun Yan**,[3] **Shuqiang Shan**,[1] **and Rongjun Wu**[1,2]

[1]*School of Mathematics, Southwest Minzu University, Chengdu, Sichuan 610 041, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541 004, China*
[3]*Faculty Affairs Office, Southwest Minzu University, Chengdu, Sichuan 610 041, China*

Correspondence should be addressed to Xiaoguang Liu; dtcr-gg@163.com

Received 2 December 2021; Revised 30 December 2021; Accepted 10 January 2022; Published 1 February 2022

Academic Editor: Thippa Reddy G

Electronic medical records (EMR) have been commonly used in medical institutions in recent years. In particular, the combination of EMR and the cloud server has significantly improved the work efficiency and therapeutic level of the hospital. It also raises some security concerns, e.g., the information leaks. Blockchain has features including decentralization, traceability, openness, and tamper resistance. Therefore, the technology may be used to overcome the above flaws. In this paper, we introduce a new blockchain-assisted EMR in the cloud environment by using proxy reencryption and sequential multisignature. Firstly, blockchain makes the scheme have high-security performance without a trusty center. Secondly, we use proxy reencryption to protect personal medical data while helping doctors to access patients' historical medical records. Moreover, the doctors have used a sequential multisignature, which is practical and can effectively improve security performance. The analysis results show that the proposed scheme can satisfy various security features of EMR and has an ideal computational and communication cost. Finally, the scheme is implemented to show its performance.

1. Introduction

With the full application of modern information technologies such as big data, cloud computing, and artificial intelligence in the medical field, medical informatization has exerted a significant influence on the optimal allocation of medical resources [1, 2]. EMR has emerged from this context, and it uses electronic devices (such as computers and smartphones) to store, manage, and transmit digitized medical records [3]. It can significantly enhance the work efficiency and therapeutic level of the hospital [4]. Also, EMR provides a judgment basis for dealing with medical malpractice [5]. When a patient goes to see a doctor, his/her medical history can help the doctor make an accurate diagnosis. However, most patients are often unable to detail their medical history due to long-time intervals and a lack of relevant expertise. It will affect the current diagnosis and increase the fiscal burden. Therefore, an ideal EMR should be able to help doctors timely obtain complete and accurate historical medical information. Furthermore, security and privacy preservation are crucial in EMR since medical information is sensitive and personal [6, 7].

EMR has developed significantly in recent years of its remarkable advantages, such as transmitting fast and easy to use [8, 9]. Notably, the emergence of cloud storage is a new milestone in the development of EMR [10]. They move medical data from the traditional data center to a cheaper and safer cloud server. It can improve work efficiency and allow hospitals to invest more time and resources in diagnosis and care. Thus, researchers proposed many cloud-assisted EMR architectures in recent years. However, the privacy, confidentiality, and integrity of medical data will face more threats since the data is outsourced to a third party, i.e., the cloud [11]. For example, doctors can collude with the cloud server to modify their erroneous diagnoses in medical malpractice. So, how to improve the efficiency of

data storage and data sharing while ensuring data security and protecting patient privacy is the focus of research [12, 13]. It is necessary to design a lightweight, efficient, and secure EMR system.

Blockchain technology was introduced in 2008 [14]. It is a decentralized distributed (distributed in multiple locations and able to work together) database system. Blockchain has features of decentralization, tamper-resistant, openness, autonomy, and traceability. It can effectively overcome the adverse effects of centralization and reduce the cost of trust [15]. Therefore, blockchain may be a promising assisted technology of EMR, and it has received attention [16]. However, when blockchain technology is applied to the medical industry, it must be measured between improving efficiency and reducing cost. Only when appropriate blockchain technology is adopted and the system efficiency and operating cost are well balanced, can the business model be established. In addition, many problems are still unsolved before satisfying the practical application in recurrent [17, 18]. For example, (1) the data owner usually encrypts the data with the public key of the user or the session key of both parties, which leads to weak data sharing; (2) only the diagnosis of a single doctor was considered, regardless of a situation in which multiple doctors consult; (3) the cost of computing, communication, and storage is too high.

In this paper, we propose a blockchain-assisted EMR in the cloud environment by using proxy reencryption and sequential multisignature, and we call it BC-EMR. In BC-EMR, a group key and a sequential multisignature are utilized to enhance data security (a diagnosis may be made by a doctor or multiple doctors) [19]. Proxy reencryption helps doctors to access patients' historical medical records while protecting data [20]. Especially, blockchain technology has enabled BC-EMR to overcome many flaws in general cloud-assisted EMR and dramatically improve security [21]. BC-EMR has an ideal computational and communication cost. The main contributions are listed as follows:

We establish a group key between a hospital's server and the doctors of one team utilizing a lightweight one-to-many authentication protocol. It can protect the patient's information.

We propose a blockchain-assisted EMR in the cloud environment by using proxy reencryption and sequential multisignature. The proposed scheme not only can realize the safe storage of data but also make secure data sharing between doctors at different hospitals.

The security analysis of BC-EMR is given. The results show that BC-EMR can satisfy various security features. It also can fend off some specific threats, such as illegal cooperation between doctors and the cloud. Finally, we compare the computational and communication cost of BC-EMR with three existing schemes and then have implemented BC-EMR.

The remainder of the paper is organized as follows. In Section 2, we introduce related works. The preliminaries are presented in Section 3. In Section 4, we introduce the details of BC-EMR. In Section 5, the security analysis of BC-EMR is given. In Section 6, we evaluate the performance of BC-EMR and implement it. Finally, we conclude our paper in Section 7.

2. Related Works

Ekblaw et al. [22] used the Ethereum platform to realize MedRec that is a medical information sharing platform combining medical blockchain and big data. The system makes use of blockchain, the embedded authentication system, the security system, and an accountability system, which can provide users with powerful security technology when dealing with sensitive information. Xia et al. [23] proposed a blockchain-based health data sharing architecture that only allows the invited (verified) users to access. Thus, it solves many of the access control challenges associated with sensitive data. They have also come up with a system called MeDShare in [24]. The scheme deals with the problem of sharing medical data with big data custodians in untrusted environments. It uses smart contracts and access control mechanisms to track data behavior. Xue et al. [25] proposed a medical blockchain system by combining the medical server and auditing server. Zhang et al. [26] used a hospital-owned private blockchain to store patients' health data, and the consortium blockchain to store safety indexes for personal health data. In particular, the authors have described the details and implemented the scheme on JUICE. In [27], Ivan analyzed the feasibility of using blockchain to protect health data, the implementation barriers, and specific plans for transitioning from current technology to blockchain solutions. Cao et al. [28] proposed a secure cloud-assisted EMR. This scheme utilizes Ethereum platform-based blockchain to protect outsourced medical data. Because every operation of the EMR is put into the blockchain as a transaction, it has excellent security. Esposito et al. [29] comprehensively analyzed the potential of blockchain to protect medical data in the cloud. They also pointed out the practical challenges and future works. Israa et al. [30] elaborated on the benefits and threats of blockchain technology in healthcare. Abdellatif et al. [31] introduced a new smart and safe healthcare system, which takes advantage of edge computing and blockchain to allow for epidemic detection and remote monitoring. The system also allows for the secure exchange of medical data between local medical entities. Shen et al. [32] analyzed the topological relationship among participants in the process of income distribution and established some Shapley value models from simple to complex. Based on the analysis of distribution rules, the incentive effect of secure data sharing and the rationality of the design scheme is discussed. Patil et al. [33] proposed an efficient blockchain authentication protocol for the Internet of Things based on the secret computational model of a physically unclonable function, which can guarantee data provenance and data integrity. Based on the elliptic curve digital signature algorithm, Xiong et al. [34] introduced an efficient and large-scale batch verification scheme with group testing technology for blockchain-enabled IoMT. Zhang et al. [35] proposed a reliable and efficient system based on edge computing and blockchain.

Simulation results show that the proposed method has better computational efficiency and higher reliability than the existing methods. Cheng et al. [36] designed a blockchain-based data-sharing network model for medical cyber-physical systems and used BAN logic to analyze security protocols. Saini et al. [37] built an access control framework based on smart contracts, which is built on top of distributed ledger (blockchain) to ensure EMR sharing between different entities involved in smart healthcare systems.

In Table 1, we give a comparison between the different schemes introduced above. For convenience, we let F1, F2, F3, F4, and F5 denote payment for the blockchain platform, consensus mechanism and reduce the pressure for the main chain, the demand for calculating power, and the private blockchain.

3. Preliminaries

3.1. Blockchain. Blockchain is a novel application of distributed data storage, peer-to-peer transmission, and consensus mechanism, etc. [38]. As shown in Figure 1, a blockchain system consists of many blocks, and each block contains a block header and a block body. The block header includes the hash value of the current block, the timestamp, the hash value of the previous block, and so forth. Block body stores some transaction records. Its main characteristics are listed as follows:

(1) Decentralization: there are numerous nodes distributed in the blockchain network, which can be freely connected to exchange information without any third institution.

(2) Tamper resistance: after the information is added to the blockchain by consensus mechanism, all nodes will record it. Each block contains the hash value of the previous block. If a block's data is modified, all the blocks behind that block need to be changed, which is almost impossible.

(3) Traceability: blockchain stores all data through the block data structure, and any data stored in the blockchain can trace its origin through the chain structure.

(4) Openness: any node can get the ledger of the whole network. Except for the information of the parties directly related to the data being encrypted by asymmetric encryption technology, other data is open to all nodes.

(5) Autonomy: the use of a consensus mechanism in blockchain enables all nodes in the whole system to freely and securely exchange, record, and update data.

3.2. The Basic Requirements of EMR

(1) *Security and Privacy Preservation*. (a) The system can resist malicious attacks on medical data such as forgery attacks, modification attacks, replay attacks, guess attacks, man-in-the-middle attacks, and trackable attacks. (b) *Nonrepudiation*. The participants cannot deny the historical data generated by themselves. (c) *Confidentiality*. The data transmitted and stored in the network is sensitive personal information, so the system needs to resist data leakage. (d) *Authenticity*. The data cannot be illegally modified. For example, in the cloud environment, the system can prevent doctors and the cloud from conspiring to tamper with medical data.

(2) *Data Sharing*. Authorized third parties such as other doctors can access the patient's historical medical data with the consent of the patient. In particular, these data may be generated from different doctors at different hospitals.

(3) *Patient Control*. Patients can control other people's access to their historical medical records.

(4) *Uniform Standard*. There are uniform data standards and sharing principles among all participants in the system, which is conducive to improving the efficiency and stability of the system.

3.3. Bilinear Map. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ denote two multiplicative groups, respectively, and they have the same prime order p. If a map e: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ satisfies the following three properties, then e is called the bilinear map [39]:

(1) Bilinear: for any points $A, B \in \mathbb{G}_1$ and any points $a, b \in \mathbb{Z}_p^*$, $e(A^a, B^b) = e(A, B)^{ab}$ is satisfied.

(2) Nondegeneracy: there is a point $A \in \mathbb{G}_1$ so that $e(A, A) \neq 1$, 1 is $\mathbb{G}_2$'s an identity element.

(3) Computability: for any points $A, B \in \mathbb{G}_1$, $e(A, B)$ can be computed within polynomial time.

3.4. Intractable Problems

(1) Discrete Logarithm Problem (DLP): knowing two points A and B in $\mathbb{G}_1$ and $A = B^n$, it is hard to find $n \in \mathbb{Z}_p^*$ so that $A = B^n$.

(2) Computational Diffie–Hellman Problem (CDH): knowing point A in $\mathbb{G}_1$, for a given (A, A^m, A^n), it is hard to compute A^{mn}, where $m, n \in \mathbb{Z}_p^*$.

3.5. Proxy Reencryption. Proxy reencryption means that a delegate A generates a proxy reencryption key $PK_{A \longrightarrow B}$ of a delegatee B and then sends $PK_{A \longrightarrow B}$ to the agent. The agent uses $PK_{A \longrightarrow B}$ to convert the ciphertext encrypted with A 's public key PK_A to the ciphertext encrypted with B 's public key PK_B. It does not need to use A 's private key to decrypt the ciphertext, and we will list the details as follows [40]:

(1) A encrypts the plaintext M with PK_A, i.e., $C_A = E_A(PK_A, M)$.

(2) A generates the proxy reencryption key $RK_{A \longrightarrow B}$ for B and sends C_A and $RK_{A \longrightarrow B}$ to the agent.

(3) The agent converts C_A into C_B utilizing $RK_{A \longrightarrow B}$, where C_B is M's ciphertext encrypted with PK_B.

Table 1: Comparison between existing schemes.

Schemes	F1	F2	F3	F4	F5
[22]	√	POW	×	Big	×
[23]	×	DPOS	√	Small	√
[24]	×	DPOS	√	Small	√
[25]	×	Improved DPOS	√	Small	√
[26]	×	DBFT	√	Big	√
[28]	√	Improved DPOS	√	Small	√
[36]	×	Improved DPOS	√	Small	×
[37]	×	POW	√	Small	×
Ours	×	Improved DPOS	√	Small	√

√Support; ×not-support.

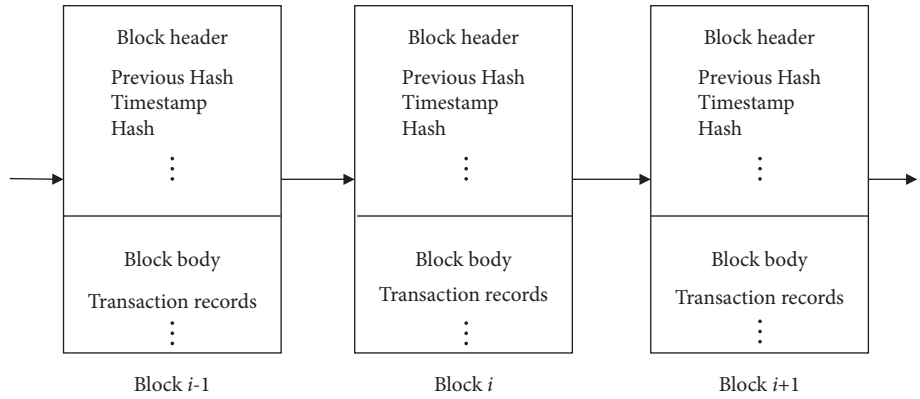


Figure 1: The basic structure of blockchain.

Notably, the agent only makes the transformation service of ciphertext and does not know M.

(4) The agent sends C_B to B that decrypts it using its private key to get M.

3.6. Sequential Multisignature. Sequential multisignature is a particular digital signature scheme. It means that multiple users sign the message in a specific order [19, 41]. A general sequential multisignature usually needs to execute the following four algorithms, i.e., Setup, Key Generation, Sign, and Verify:

(1) Setup: the key generation center (KGC) inputs a security parameter and generates system parameter para, and system master key.

(2) Key Generation: given para, users $N_i\,(i = 1, 2, \ldots, n)$ generate their private key SK_i and then compute their own public key PK_i by inputting SK_i.

(3) Sign: the signer $N_i\,(i = 2, \ldots, n)$ orderly verifies the partial signature s_{i-1} of the previous signer N_{i-1}. If it is valid, N_i outputs own partial signature s_i signed by SK_i.

(4) Verify: the verifier verifies the signatures by inputting $(m, ID_i, PK_i, s_n)\,(i = 1, 2, \ldots, n)$ and the order of signature.

4. The Proposed BC-EMR

4.1. System Model. In this section, the details of BC-EMR will be given. We use the sequential multisignature of [41] and the proxy reencryption of [42] to construct the scheme. As shown in Figure 2, BC-EMR mainly consists of four entities, i.e., a doctor team, a patient, a hospital, and a cloud server. In BC-EMR, a patient first registers at the hospital. If the identity is approved, the hospital server assigns a medical team to the patient based on the initial condition. Then, the server and members of the team establish a group key that is used to protect the patient's diagnosis results. In the diagnosis, when a doctor receives a message from the former doctor, he/she first verifies previous all doctors' signatures. If it passes, the doctor will make the diagnosis and broadcast his/her signature in the blockchain. Otherwise, he/she requests the former doctor to resend the message. When the last doctor has finished the signature, he/she encrypts the result by using the patient's public key and sends the ciphertext to the cloud server. The ciphertext and signatures will be stored in the cloud and blockchain, respectively, if the signatures pass the verification. Different doctors at different hospitals have the right to access the patient's medical history with the patient's consent. BC-EMR includes the following five phases, i.e., Initialization, Group key generation, Diagnose, Data storage, and Data sharing. In Table 2, we give the used notations in the paper.

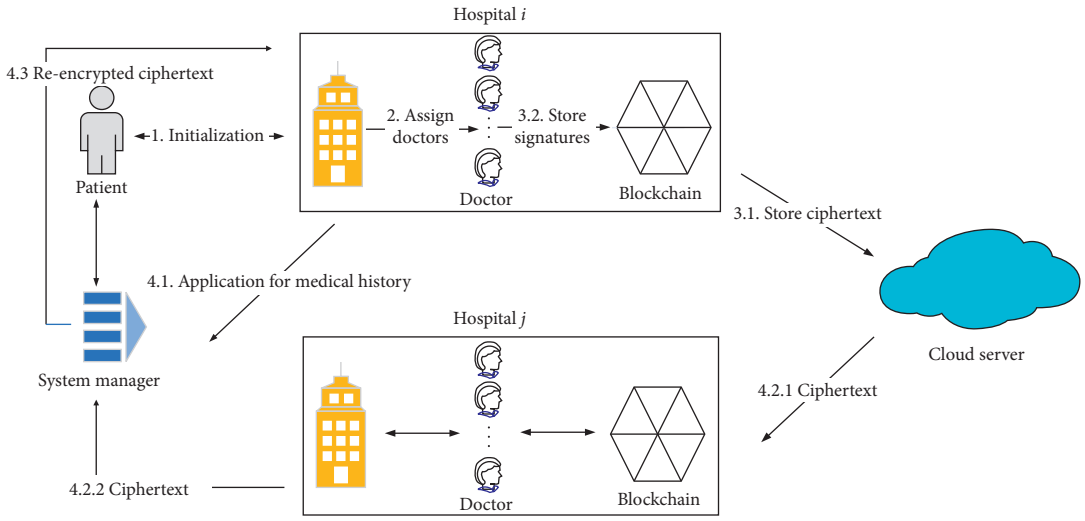


Figure 2: The basic structure of BC-EMR.

Table 2: Notations.

Notation	
p, q	Two prime numbers
SM	The system manager
H_i	The ith hospital
P_j	The jth patient
D_k	The kth doctor
$PK_{(.)}$	The public key
$SK_{(.)}$	The private key
$ID_{(.)}$	The identity
s	The diagnosis order
g	The generator of $\mathbb{G}_1$
KGC	The key generation center
$E_{(.)}$	Encryption
$D(.)$	Decryption
e	The bilinear map
$H_{(.)}$	The hash function
MAC	The message authentication code
γ	The security parameter

4.2. Initialization

(1) SM inputs a security parameter 1^γ, selects the bilinear map e: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ and a random number $\overline{g} \in \mathbb{G}_1$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative groups with the same prime order p. g is a generator of $\mathbb{G}_1$. Four hash functions are defined as follows: H_0: $\{0,1\}^* \longrightarrow \mathbb{G}_1$, H_1: $\{0,1\}^{\leq l} \longrightarrow \mathbb{G}_1$, H_2: $\{0,1\}^{\leq l} \longrightarrow \mathbb{G}_1$, and H_3: $\mathbb{G}_2 \longrightarrow \{0,1\}^\gamma$, where l is the length of the verification keys [42]. Besides, SM selects a random number $x \in \mathbb{Z}_p^*$ as the system master key, and the public key $Y = g^x$. The public parameters of BC-EMR are $\{p, g, Y, \overline{g}, H_0, H_1, H_2, H_3, e, \mathbb{G}_1, \mathbb{G}_2\}$. In BC-EMR, we limit the number of the signer reissues the signature to no more than N.

(2) Hospital H_i selects a random number $h_i \in \mathbb{Z}_p^*$ as its private key and the public key $PK_i = g^{h_i}$.

(3) Patient P_j selects a random number $p_j \in \mathbb{Z}_p^*$ as the private key and sets $PK_j = g^{p_j}$ as the public key.

(4) Doctor D_k randomly selects $d_k^1, d_k^2, d_k^3 \in \mathbb{Z}_p^*$, computes $A_k = g^{d_k^1}$, $B_k = g^{d_k^2}$, and $C_k = g^{d_k^3}$. The private key is (d_k^1, d_k^2, d_k^3) and the public key $PK_k = (A_k, B_k, C_k)$.

4.3. Group Key Generation.

When a patient P_j sees a doctor in the hospital H_i, P_j sends an identity ID_j and symptoms to HO_i 's server securely. If the identity is legal, the server first selects a random number $\lambda_j \in \mathbb{Z}_p^*$, computes P_j 's pseudoidentity $PID_j = E_{h_i}(ID_j \oplus \lambda_j \| \lambda_j)$ and sends it to P_j. It also assigns initial doctors $D_k (k = 1, \ldots, n)$ to make a diagnosis according P_j 's condition, sends the evidence $\alpha \in \{0,1\}^*$ and a diagnosis order s to P_j, and sends a signature timestamp T, α, and s to $D_k (k = 1, \ldots, n)$ securely. Especially, as in Figure 3, a group key between H_i and $D_k (k = 1, \ldots, n)$ will be set to protect medical information. The details are given as follows:

(1) H_i chooses a random number $l_i \in \mathbb{Z}_p^*$, computes $U_i = g^{l_i}$, and sends (ID_i, U_i) to D_k.

(2) D_k randomly selects a number $l_k \in \mathbb{Z}_p^*$, computes $V_k = g^{l_k}$, and sends (ID_k, V_k) to H_i.

(3) H_i computes $s_i = V_k^{l_i}$, $MAC_i = MAC_{s_i}(ID_k, V_k, U_i)$, and sends MAC_i to D_k.

(4) D_k computes $s_k = U_i^{l_k}$ and $MAC_k = MAC_{s_k}(ID_k, V_k, U_i)$. If $MAC_i = MAC_k$, D_k computes $MAC_k^\dagger = MAC_{s_k}(ID_i, U_i, V_k, s_k)$ and sends $MAC_k^\dagger$ to H_i. Otherwise, $\perp$.

(5) H_i computes $MAC_i^\dagger = MAC_{s_i}(ID_i, U_i, V_k, s_i)$ and checks $MAC_i^\dagger = MAC_k^\dagger$. If not, $\perp$. Otherwise, H_i computes $K = V_1^{l_i} \ldots V_n^{l_i}$ and $M = E_{s_i}(K)$, and then sends M to D_k.

(6) D_k decrypts M using s_k to get the group key K.

The correctness of the above protocol is based on the following equation

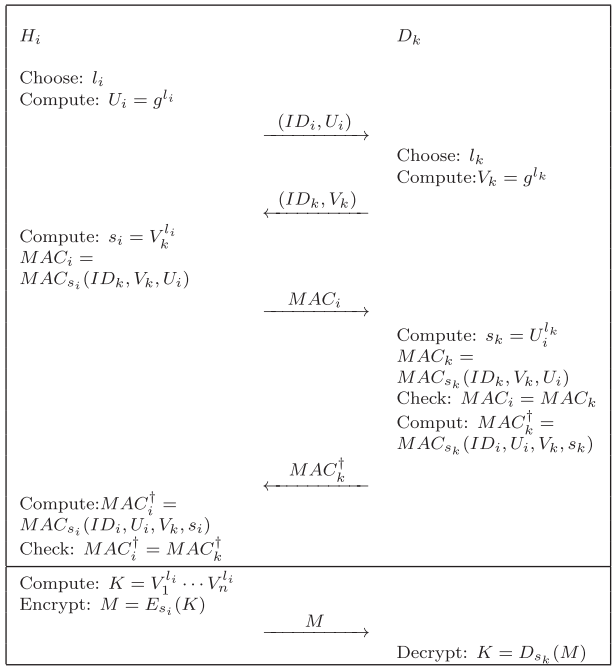


Figure 3: The group key generation.

$$s_i = V_k^{l_i} = g^{l_i l_k} = U_i^{l_k} = s_k. \tag{1}$$

4.4. Diagnosis

(1) P_j shows α and PID_j to D_1 as the evidence, so that D_1 makes a diagnosis or accesses the history records of P_j. If it is legal, D_1 first generates a diagnosis m_1, randomly selects $r_1 \in \mathbb{Z}_p^*$, computes $R_1 = g^{r_1}$, $X_1 = R_1^{d_1^2 + d_1^3}$, $W_1 = H_0(m_1, T)$, $Q_1 = W_1^{d_1^1} X_1$ and $c_1 = E_K(m_1)$. Then D_1 sends the signature message $(c_1, PID_j, (R_1, Q_1))$ to D_2. Meanwhile, D_1 broadcasts signature (PK_1, PID_j, R_1, Q_1) in the blockchain; please see Figure 4 for the structure of block. In BC-EMR, each block is used to store one patient's information such as all doctors' signatures.

(2) P_j shows α to $D_k (k = 2, \ldots, n)$. If it is legal, D_k confirms whether he/she received $(c_{k-1}, PID_j, (R_{k-1}, Q_{k-1}))$ before $T_k = kT$. If not, D_k requests D_{k-1} to resend the message. Then, D_k decrypts c_{k-1} to get $m_1, \ldots, m_{k-1}$ and verifies the following:

$$e(Q_{k-1}, g) = \prod_{i=1}^{k-1} e(W_i, A_i) e\left(\prod_{i=1}^{k-1} B_i C_i^i, R_{k-1}\right). \tag{2}$$

If it is true, D_k first randomly selects $r_k \in \mathbb{Z}_p^*$, generates diagnosis m_k, computes $R_k = R_{k-1} g^{r_k}$, $X_k = R_k^{d_k^2 + k d_k^3}$, $Z_k = (\prod_{i=1}^{k-1} B_i C_i^i)^{r_k}$, $W_k = H_0(m_1 \| m_2 \ldots \| m_k, T)$, and $Q_k = W_k^{d_k^1} Q_{k-1} X_k Z_k$. Then D_k encrypts the results $m_1 \| m_2 \ldots \| m_k$ as $c_k = E_K(m_1 \| m_2 \ldots \| m_k)$ and sends the signature $(c_k, PID_j, (R_k, Q_k))$ to D_{k+1}. Meanwhile, D_k boardcasts signature (PK_k, PID_j, R_k, Q_k) in the blockchain. Thus, the final diagnosis is $m = m_1 \| m_2 \ldots \| m_n$ and the signature message is $(c_n, PID_j, (R_n, Q_n))$.

(3) D_n encrypts the results m using P_j's public key PK_j to generate the ciphertext C_j. We will give the details as follows:

(a) D_n selects a general signature key pair (PK, SK) and sets $PK = A$

(b) D_n randomly selects a number $r \in \mathbb{Z}_p^*$ and computes $B = PK_j^r$, $C = e(g, H_1(A))^r \oplus m$, $D = H_2(A)^r$, $E = \overline{g}^r$, $F = e(PK_j, H_0(\alpha))^r$, and $G = H_3(F)$

(c) D_n signs the message (C, D, E, G) using SK and outputs the ciphertext $C_j = (S, A, B, C, D, E, G)$, where S is the signature

(d) D_n sends ciphertext C_j and PID_j to the cloud server

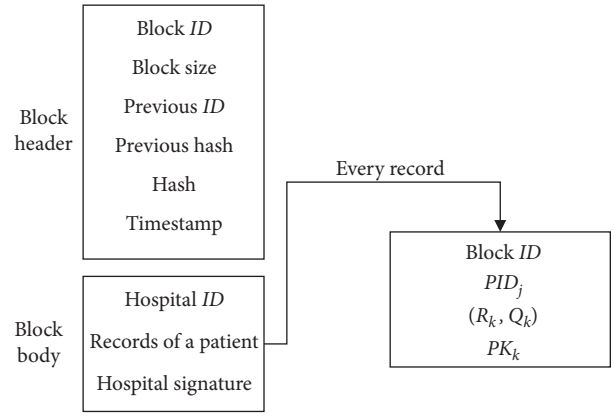


Figure 4: The structure of block in the hospital's blockchain.

4.5. Data Storage. In BC-EMR, every doctor is the general node of the blockchain. The cloud server and HO_i's server are the supernodes, and they are responsible for verifying the signature message. That is, if the signature message passes their verification, all nodes will put the current signatures about the patient P_j in a block and update their stored records. The verification scheme is that the supernodes check the following equation:

$$e(Q_n, g) = \prod_{i=1}^{n} e(W_i, A_i) e\left(\prod_{i=1}^{n} B_i C_i^i, R_n\right). \tag{3}$$

If it is true, the supernodes send a confirmation message in the blockchain so that all nodes accept the signatures about P_j, put them in a block, and update the stored records. Otherwise, $\perp$.

4.6. Data Sharing. When the doctor D_k in the hospital H_d makes a diagnosis for the patient P_j, P_j's medical history in other hospitals H_i may help D_k. Therefore, if D_k wants, he/she can obtain these records with the consent of P_j. The details are as follows:

(1) D_k and P_j send their identities and request to SM. If it is passed, SM sends a notice to H_i, and the server of H_i extracts the ciphertext $C_j = (S, A, B, C, D, E, G)$ from the cloud server and sends it to SM. In addition, P_j sends a trapdoor $T_\alpha = H_0(\alpha)^{p_j}$ to D_k.

(2) D_k and P_j send the private keys d_k^1 and p_j to SM, respectively. Then, SM outputs the reencryption key $rk_{j\leftrightarrow k} = d_k^1/p_j$.

(3) SM checks the signature S on (C, D, E, G), i.e., $e(B, H_2(A)) = e(PK_j, D)$, and $e(B, \overline{g}) = e(PK_j, E)$. If any of them fails, $\perp$. Otherwise, SM computes $B\prime = B^{rk_{j\leftrightarrow k}} = PK_j^{rk_{j\leftrightarrow k}\times r} = (g^{p_j r})^{d_k^1/p_j} = g^{d_k^1 r} = A_k^r$ and sends the ciphertext $(S, A, B\prime, C, D, E, G)$ to D_k.

(4) D_k checks the signature S, i.e., $e(B\prime, H_2(A)) = e(A_k, D)$, $e(B\prime, \overline{g}) = e(A_k, E)$, and $G = H_3(e(B\prime, T_\alpha)^{1/d_k^1})$. If any of them fails, $\perp$. Otherwise, D_k recovers the message $m = C \oplus e(B\prime, H_1(A))^{1/d_k^1}$.

5. Solutions to the Basic Requirements

BC-EMR has provided for the advantages described in Subsection 3.1 since it uses blockchain. Especially, the scheme is based on the sequential multisignature of [41] and the proxy reencryption of [42]. They are proven secure in the random oracle model, which is based on the hardness of the CDH problem and the modified DBDH problem, respectively, and please see [41, 42] for the full formal proof.

In this subsection, we will show why BC-EMR satisfies the basic requirements of BMR. In Table 3, we list the comparison results about BC-EMR and the other three blockchain-based EMR schemes ZL, CZ, and AS. Here, the schemes in [26, 28, 37] are denoted as ZL, CZ, and AS, respectively.

5.1. Security and Privacy Preservation (SP)

(a) Malicious attacks (MA)

Forgery attack (M1): to get the doctor's private key to generate a legal signature, the adversary must solve DLP intractable problem. Especially, it is not feasible to falsify the diagnosis by the cloud server or the collaboration between the patient and the cloud server. The reason is that they also can not get the doctors' legal signatures from the hospital's server or SM can detect any forged information by verifying the signatures stored in the blockchain. So, BC-EMR can resist the forgery attack.

Mo di ficationattack (M2): in BC-EMR, the last doctor encrypts the diagnosis results with the patient's public key and outsources them to the cloud. If an adversary wants to modify them, it first needs to obtain the patient's private key and the cloud's permission. To get the private key illegally, the adversary must solve DLP intractable problem. Therefore, it is not possible. More importantly, BC-EMR stores the doctors' signatures of their diagnoses in the blockchain. Then, it is easy to detect the modification to the diagnosis results, even if the patient's private key is leaked or the patient (or doctor) cooperates with the cloud. Thus, BC-EMR can resist the modification attack.

Replayattack (M3): BC-EMR has introduced the timestamp T. D_k will confirm whether he/she received the message before $T_k = kT$. It is impossible to change timestamp T since the signatures stored in the blockchain contain it. Any modification to T is easily detected, and thus BC-EMR can resist the replay attack.

Guessattack (M4): in BC-EMR, the system sets the number of resigning as N. If the number of resigning exceeds N, the signature terminates. It can be to limit the number of attacks effectively and resist the guessing attack.

Man − in − the − mi dd leattack (M5): protection against the man-in-the-middle attack follows from the protection against the forgery attack, modification attack, and replay attack.

Trackableattack (M6): the patient and hospital generate different random numbers including l_i, l_k, and r_i in each execution of BC-EMR. Thus, there is no constant value in the transmitted or stored messages, and the adversary can not trace the action. Therefore, our BC-EMR can resist the trackable attack.

(b) Nonrepudiation (NR): blockchain technology makes BC-EMR satisfy traceability. We can search the origin of any record stored in the blockchain by the chain structure and the doctor's signature. Thus, no participant can deny the data generated by himself/herself.

(c) Confidentiality (CO): before diagnosis, the hospital server will set a pseudoidentity for each patient. During the diagnosis, the patient will use the pseudoidentity to interact with doctors. The doctors will encrypt the diagnosis results with the group key, and only members of the medical team can get them. When the diagnosis is over, the last doctor will encrypt the result with the patient's public key before storing it in a cloud server. If the adversary wants to get the patient's private key to decrypt the ciphertext, he/she needs to solve the DLP intractable problem. So BC-EMR has ideal confidentiality.

(d) Authenticity (AU): no one but the patient can decrypt the ciphertext of diagnosis since the final

Table 3: Comparison of the basic requirements.

	SP									DS	PC	US
	M1	M2	M3	M4	M5	M6	NR	CO	AU			
ZL	√	√	√	×	√	√	√	√	√	√	√	√
CZ	√	√	√	×	√	√	√	√	√	×	×	√
AS	√	√	√	×	√	√	√	√	√	√	√	√
BC-EMR	√	√	√	√	√	√	√	√	√	√	√	√

√Support; ×not-support.

doctor encrypts diagnosis results with the patient's public key. Doctors have stored the signatures of diagnoses in the blockchain, and these violations are easy to spot.

5.2. Data Sharing (DS). The scheme has utilized proxy reencryption technology. If the doctor has obtained the consent of the patient, he/she will get the ciphertext encrypted by their public key. Then the doctor accesses the patient's historical medical records by decrypting the ciphertext. That is, BC-EMR realizes data sharing between different doctors at different hospitals.

5.3. Patient Control (PC). When a doctor needs to know the patient's historical records of another hospital, the doctor must get the ciphertext that is encrypted by his/her public key. However, the original ciphertext is encrypted by the patient's public key. The transformation of ciphertext needs to be completed by *SM* using the reencryption key that is computed by *SM* utilizing the patient's private key. So, the patient can control the doctor to access the historical data.

5.4. Uniform Standard (US). In hospitals' BC-EMR systems, we can use a uniform standard such as the same encryption algorithm. It is beneficial to implement data sharing and other functions.

In Table 3, we give the comparison results of BC-EMR, ZL, CZ, and AS according to the basic requirements. We can know that ZL and AS can not resist the guess attack. CZ not only can not resist the guess attack but also can not satisfy patient control and make the essential data sharing.

Remarks. Without the blockchain, if a doctor tries to forge EMR that has been outsourced to a cloud server in a medical accident, he/she can incentivize the cloud server to forge or modify the existing EMR at will. This is consistent with reality. The introduction of blockchain makes the scheme resistant to threats such as doctor-cloud collusion to forge or modify EMR without additional security mechanisms, strong hypotheses, and trusted entities. In other words, blockchain plays a key role in ensuring security in BC-EMR. Moreover, in BC-EMR, blockchain only stores some lightweight information such as the doctors' signatures, and diagnostic results are stored in the cloud, thus reducing the burden of blockchain and facilitating the future implementation of the scheme.

6. Performance Evaluation

In this section, we will evaluate BC-EMR from the following two aspects: (1) computational and communication cost; (2) the implementation of BC-EMR.

6.1. Computational and Communication Cost. In this subsection, we will compare the main computational cost and communication cost of BC-EMR, ZL, CZ, and AS. *SM* usually has sufficient computational power and storage capacity, so we consider the burden on the patient and the last doctor of the team (the last doctor is responsible for outsourcing data, he/she needs to pay for more cost than other doctors). The comparison results of computational cost are shown in Table 4. Here, m denotes the scale multiplication operator in $\mathbb{G}_1$, e denotes the exponentiation operation in $\mathbb{G}_1$, b denotes the bilinear pairing operation. We ignore the remaining operations because of their low computational cost.

In Table 4, λ is the size of the disease keyword set [26]. On the doctor's side, ZL has the highest computational cost of $(17 + \lambda)m + 7e + 4b$ since λ is usually large such as 1000 in ZL. The cost of CZ is lower than BC-EMR, but it can not satisfy a critical feature of EMR, i.e., the data sharing between doctors. CZ also omits the authentication between the server and doctors in generating the treatment key, and the current doctor only verifies the previous doctor's signature. But BC-EMR will make the authentication in creating the group key, and the current doctor verifies previous all doctors' signatures. The cost of AS is also lower than BC-EMR on the doctor's side, but the scheme needs to decrypt the ciphertext using the patient's private key and then encrypt the EMR using the shared secret key in the process of data sharing. Every time generation and management of the shared secret key both are consuming cost and there is a risk of data leakage. So BC-EMR has higher security, and the additional computational cost is worthy. In addition, the results in [43] show that the computational cost of the operation e is approximately two times that of the operation m. So, on the side of the patient, we can find that BC-EMR has the lowest computational cost of $2e$. It is worthy to note that doctors often have relatively reliable computational power, and it is vital to reduce the computational burden on patients.

In Table 5, we will give the main communication cost of the three schemes. $|x|$ and $|t|$ denote the size of the element in the ciphertext space and the size of the timestamp, respectively. $|n_p|$ is the number of the private blockchain's verifiers in ZL, and $|D_p|$ is the size of personal data in AS. We use the supersingular curve $E(F_q)$ with order p over the finite field $\mathbb{GF}_q$. To give a more explicit comparison of communication cost, we assume the prime number p is 160 bits, the prime number q is 1024 bits, the point in $\mathbb{G}_1$ is 1024 bits, the point in $\mathbb{G}_2$ is 512 bits, the point in the ciphertext space, the hash value, and α all are 160 bits, the timestamp, the order message s, and the identity all are 32 bits, the security parameter γ is 512 bits, $n_p = 3$ in ZL, l is 512 bits, and $|D_p|$ is 1024 bits. In Figure 5, we give the comparison diagram of communication cost versus λ (without loss of generality, we assume $n = 3$). Besides, we provide a comparison diagram of communication cost versus the number of doctors n in Figure 6. In ZL, $\lambda = 1000$, but we set $\lambda = 200$ for clearly showing the comparison results in Figure 6.

We can see from Figure 5 that the communication cost of ZL linearly increases with λ. The communication cost of CZ, AS, and BC-EMR is constant versus λ. Since λ is usually relatively large such as 1000 in ZL. So, ZL has the highest communication cost. In Figure 6, the communication cost of CZ and BC-EMR both linearly increase with n, and the communication cost of BC-EMR is higher than CZ's. AS has

Table 4: Comparison of the computational cost.

Scheme	Patient	Doctor
ZL	$7m$	$(17+\lambda)m+7e+4b$
CZ	$5e+nm$	$2b$
AS	$5m$	$9m$
BC-EMR	$2e$	$(11+n)e+(n+9)b$

Table 5: Comparison of the communication cost.

Scheme	Communication cost
ZL	$(\lambda+12)\lvert\mathbb{G}_1\rvert+\lvert\mathbb{G}_2\rvert+5\lvert\mathbb{Z}_p^*\rvert+13\lfloor 2/3n_p\rfloor+\lvert t\rvert+\lvert x\rvert+\lvert I\ D\rvert+2\lvert\text{Hash}\rvert$
CZ	$(3n+1)\lvert\mathbb{G}_1\rvert+n\lvert\mathbb{Z}_p^*\rvert+2(n+1)\lvert I\ D\rvert+n\lvert\text{Hash}\rvert+2\lvert x\rvert+(n+1)\lvert t\rvert$
AS	$3\lvert\mathbb{G}_1\rvert+\lvert I\ D\rvert+4\lvert x\rvert+\lvert\text{Hash}\rvert+\lvert\mathbb{Z}_p^*\rvert+\lvert t\rvert+\lvert D_p\rvert$
BC-EMR	$(3n+8)\lvert\mathbb{G}_1\rvert+2\lvert\mathbb{G}_2\rvert+2\lvert\mathbb{Z}_p^*\rvert+5\lvert I\ D\rvert+(n+6)\lvert x\rvert+2\lvert\text{Hash}\rvert+(n+2)\lvert\alpha\rvert+n\lvert t\rvert+2\lvert\gamma\rvert+2\lvert l\rvert+(n+1)\lvert s\rvert$

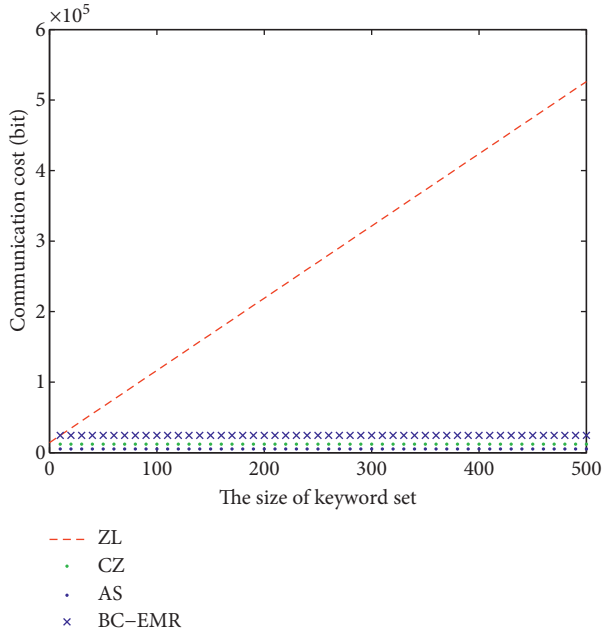


Figure 5: Communication cost comparison versus λ.

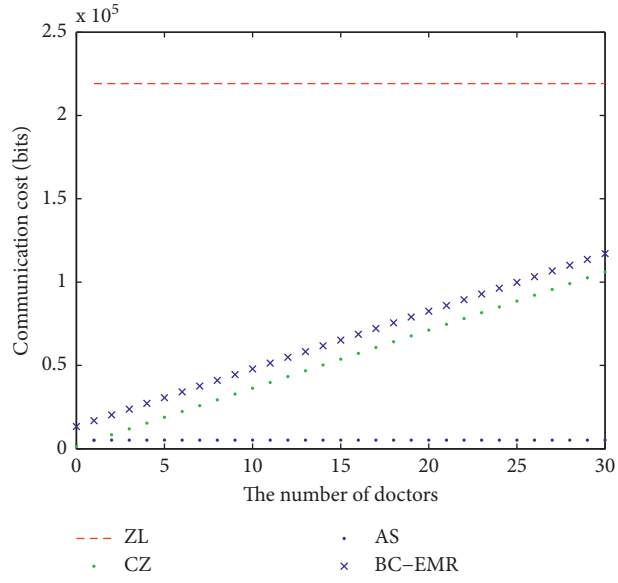


Figure 6: Communication cost comparison versus n.

the lowest communication cost, and an important reason is that the scheme does not consider the case that multiple doctors generate the EMR for a patient. In addition, as mentioned in the analysis of computational cost, CZ can not satisfy some features such as data sharing. AS requires decrypting the ciphertext using the patient's private key and then encrypting the EMR using the shared key to make data sharing. Every time the shared key is generated and managed, there is cost, and it will also increase the risk of EMR data leakage. Thus, the above results show that BC-EMR can achieve a better balance among security, basic requirements, computational cost, and communication cost. So it is an ideal EMR scheme.

6.2. Implementation of BC-EMR. In this subsection, we will give some implementation results of BC-EMR. The experiment used *Python* 3.7.4 as the programming language. Specially, we used the C language library PBC (v0.5.14) for bilinear pairing calculations, pypbc to call the PBC library in *Python*, and *Python*'s encryption algorithm library pycryptodome (v3.9.0) to implement AES encryption. We used a computer equipped with an Intel Core i7-9750H CPU @ 2.60 GHz and 15.6 GB of memory to run our experimental program. The operating system is Manjaro Linux 64 bit, the desktop environment is KDE (v5.61.0), and the kernel version is 4.19.69-1-MANJARO. During the experiment, we started multiple processes on the experimental computer. Each procedure was bound to a separate port and communicated with each other using sockets. In this way, we

Table 6: Phased time costs when $n = 3$ (ms).

Phases	Security levels		
	Level 1	Level 2	Level 3
Initialization	62.78	280.17	704.80
Group key generation	83.30	405.80	905.90
Diagnose	296.60	1366.90	2220.30
Data storage	30.10	167.14	307.40
Data sharing	193.30	971.50	2125.50

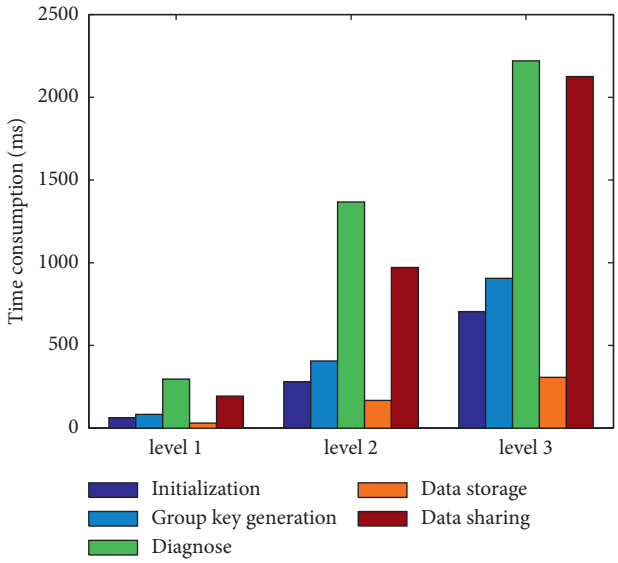


Figure 7: Phased time costs when $n = 3$.

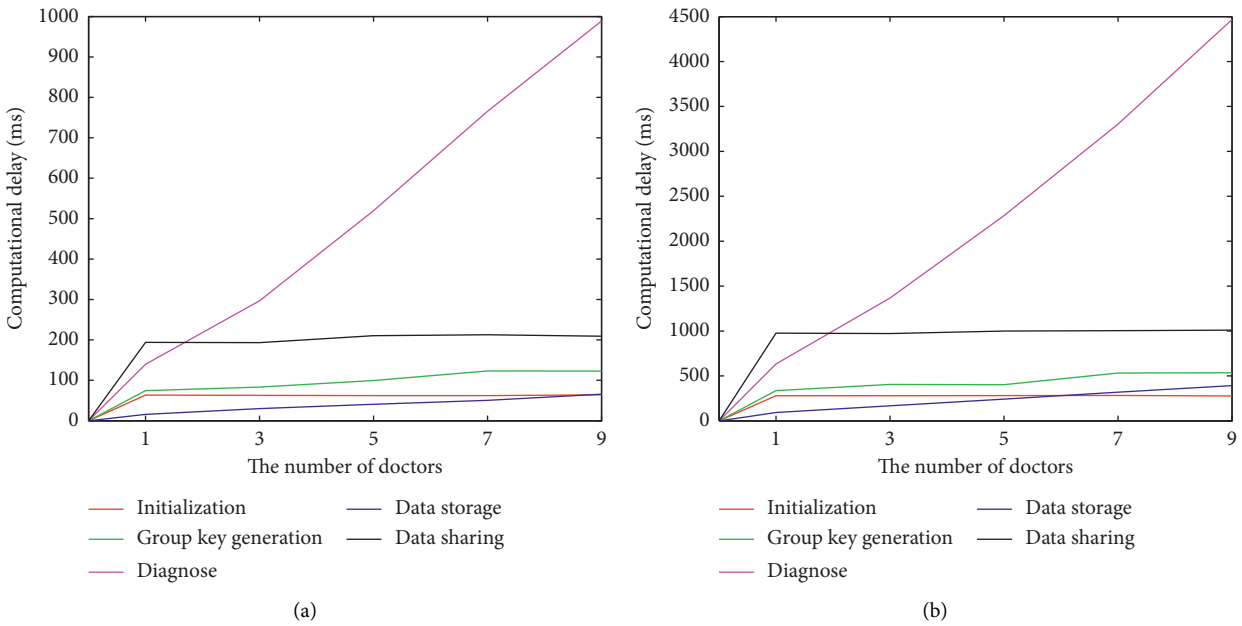
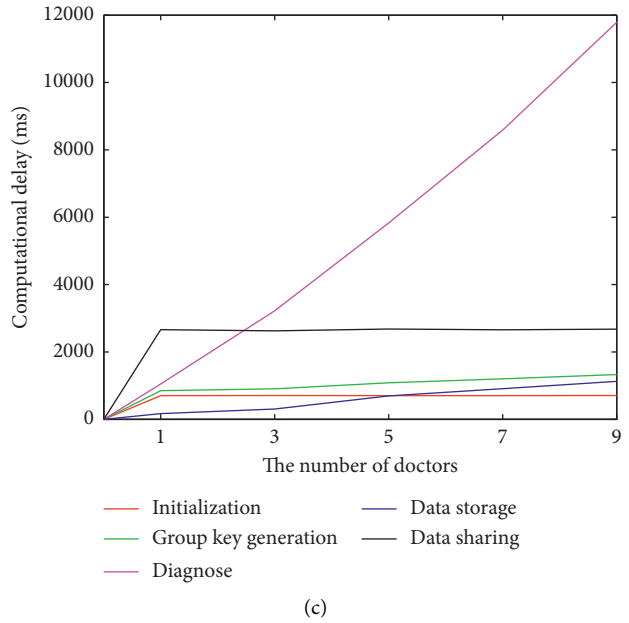


Figure 8: Continued.



(c)

Figure 8: Computational delay for three security levels. (a) Level 1. (b) Level 2. (c) Level 3.

simulated the behavior of different roles in each phase of the scenario. All values are the average result of 100 times experiments.

We consider three security levels for BC-EMR, i.e., level 1 (p = 160 bits, q = 1024 bits), level 2 (p = 224 bits, q = 2048 bits), and level 3 (p = 256 bits, q = 3072 bits). In Table 6, we assume $n = 3$ and summarize the phased computational costs for three security levels. The corresponding histogram is given in Figure 7. The result shows that the computational costs of the five stages increase as the security level increases. Besides, in Figure 8, we consider the computational delay versus n. We can find that the computational delays of the initialization phase and data sharing phase have no significant change. With the increase in the number of doctors, the other three stages' computational delays all significantly increase. The computational delay of the diagnostic phase is the fastest growing. The reason is that as the number of doctors increases, BC-EMR needs to perform more time-consuming exponential operations and bilinear pairing operations.

7. Conclusion

In this paper, we proposed a blockchain-based EMR in the cloud environment. A lightweight one-to-many authentication protocol is given to set a group key, which is used to protect the patient's diagnosis results before storing them in the cloud. The proxy reencryption is utilized to make secure data sharing between doctors at different hospitals. Blockchain and sequential multisignature technologies ensure that the stored medical information is safe. Especially, BC-EMR can resist threats such as doctor-cloud collusion to forge or modify EMR. The analysis shows that BC-EMR has a lower computational cost on the side of the patient. It is very important for EMR since the patients usually rely on resource-limited mobile devices. Besides, BC-EMR can satisfy more basic requirements and security features. So the extra computational cost on the side of the doctor and the extra communication cost compared with CZ both are worthy. That is, BC-EMR is a practical EMR. Of course, as the analysis results show, the method presented in this paper has some shortcomings, such as a slightly higher cost of communication. Since blockchain is a massive ledger backup measure, these deficiencies will directly reduce the system performance. So the balance between efficiency, security, and cost remains at the heart of what we do next.

Data Availability

Some or all data, models, or codes generated or used during the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities of Southwest Minzu University (No. 2020NQN21) and the Fund of Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS202121).

References

[1] D. V. Dimitrov, "Medical internet of things and big data in healthcare," *Healthcare Informatics Research*, vol. 22, no. 3, pp. 156–163, 2016.

[2] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, and K. Sakurai, "Authentication in mobile cloud computing: a survey," *Journal of Network and Computer Applications*, vol. 61, pp. 59–80, 2016.

[3] W. Wang, C. Qiu, Z. Yin et al., "Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, pp. 1–9, 2021.

[4] C. Y. Weng, "Data accuracy in electronic medical record documentation," *Jama Ophthalmology*, vol. 135, no. 3, pp. 232-233, 2017.

[5] S. S. Mangalmurti, L. Murtagh, and M. M. Mello, "Medical malpractice liability in the age of electronic health records," *New England Journal of Medicine*, vol. 363, no. 21, pp. 2060–2067, 2010.

[6] J. Song, Z. Han, W. Wang, J. Chen, and Y. Liu, "A new secure arrangement for privacy-preserving data collection," *Computer Standards & Interfaces*, vol. 80, Article ID 103582, 2022.

[7] F. Li, Y. Han, and C. Jin, "Cost-effective and anonymous access control for wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 747–758, 2018.

[8] R. Pivovarov, D. J. Albers, J. L. Sepulveda, and N. Elhadad, "Identifying and mitigating biases in ehr laboratory tests," *Journal of Biomedical Informatics*, vol. 51, pp. 24–34, 2014.

[9] A. Sheth, U. Jaimini, and H. Y. Yip, "How will the internet of things enable augmented personalized health?" *IEEE Intelligent Systems*, vol. 33, no. 1, pp. 89–97, 2018.

[10] J. Haskew, G. Rø, K. Saito et al., "Implementation of a cloud-based electronic medical record for maternal and child health in rural Kenya," *International Journal of Medical Informatics*, vol. 84, no. 5, pp. 349–354, 2015.

[11] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. J. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[12] V. Casola, A. Castiglione, K. R. Choo, and C. Esposito, "Healthcare-related data in the cloud: challenges and opportunities," *IEEE Cloud Computing*, vol. 3, no. 6, pp. 10–14, 2016.

[13] J. L. Fernandez-Aleman, I. C. Senor, P. A. Lozoya, and A. Toval, "Security and privacy in electronic health records: a systematic literature review," *Journal of Biomedical Informatics*, vol. 46, no. 3, pp. 541–562, 2013.

[14] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2009, http://bitcoin.org/bitcoin.pdf.

[15] W. Z. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Y. Han, and C. H. Su, "Blockchain-based reliable and efficient certificateless signature for iiot devices," *IEEE Transactions on Industrial Informatics*, pp. 1–9, 2021.

[16] F. T. Stafford and T. Horst, "Characteristics of a blockchain ecosystem for secure and sharable electronic medical records," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1340–1362, 2020.

[17] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in iot: the challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251–279, 2019.

[18] R. Z. Yang, F. R. Yu, P. B. Si, Z. X. Yang, and Y. H. Zhang, "Integrated blockchain and edge computing systems: a survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[19] Y. L. Qin and X. P. Wu, "Efficient certificateless sequential multi-signature scheme," *Journal on Communications*, vol. 34, no. 7, pp. 105–110, 2013.

[20] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[21] L. C. Wang, X. Y. Shen, J. Li, J. Shao, and Y. X. Yang, "Cryptographic primitives in blockchains," *Journal of Network and Computer Applications*, vol. 127, pp. 43–58, 2019.

[22] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: using blockchain for medical data access and permission management," in *Proceedings of the 2016 IEEE of International Conference on Open and Big Data (OBD)*, pp. 25–30, Vienna, Austria, August 2016.

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

[24] Q. Xia, E. B. Sifah, K. O. Asamoah, J. B. Gao, X. J. Du, and M. Guizani, "Medshare: trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, Article ID 14757, 2017.

[25] T. F. Xue, Q. C. Fu, C. Wang, and X. Y. Wang, "A medical data sharing model via blockchain," *Acta Automatica Sinica*, vol. 43, no. 9, pp. 1555–1562, 2017.

[26] A. Q. Zhang and X. D. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, 2018.

[27] D. Ivan, "Moving toward a blockchain-based method for the secure storage of patient records," 2018, https://www.healthit.gov/sites/default/files/9-16-drew_ivan_20160804_blockchain_for_healthcare_final.pdf.

[28] S. Cao, G. X. Zhang, P. F. Liu, X. S. Zhang, and F. Neri, "Cloud-assisted secure ehealth systems for tamper-proofing ehr via blockchain," *Informance Science*, vol. 485, pp. 427–440, 2019.

[29] C. Esposito, A. D. Santis, G. Tortora, H. Chang, and K. R. Choo, "Blockchain: a panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018.

[30] A. Israa, H. Asma, N. Anjanarani, H. Mowafa, and A. Alaa, "The benefits and threats of blockchain technology in healthcare: a scoping review,," *International Journal of Medical Informatics*, vol. 142, pp. 1–9, 2020.

[31] A. A. Abdellatif, Z. A. Abeer, M. Amr, E. Aiman, C. F. Carla, and R. Ahmed, "sshealth: toward secure, blockchain-enabled healthcare systems," *IEEE Network*, vol. 34, no. 4, pp. 312–319, 2020.

[32] M. Shen, J. X. Duan, L. H. Zhu, J. Zhang, X. J. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1229–1241, 2020.

[33] A. S. Patil, R. Hamza, H. Yan, A. Hassan, and J. Li, "Blockchain-puf-based secure authentication protocol for internet of things," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 331–338, New York, USA, January 2020.

[34] H. Xiong, C. J. Jin, M. Alazab et al., "On the design of blockchain-based ecdsa with fault-tolerant batch verification protocol for blockchain-enabled iomt," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[35] L. J. Zhang, Y. F. Zou, W. Z. Wang, Z. L. Jin, Y. S. Sue, and H. L. Chen, "Resource allocation and trust computing for blockchain-enabled edge computing system," *Computers & Security*, vol. 105, 2021.

[36] X. Cheng, F. L. Chen, D. Xie, H. Sun, and C. Huang, "Design of a secure medical data sharing scheme based on blockchain," *Journal of Medical Systems*, vol. 44, no. 52, pp. 1–11, 2020.

[37] A. Saini, Q. Y. Zhu, N. Singh, Y. Xiang, L. X. Gao, and Y. S. Zhang, "A smart-contract-based access control framework for cloud smart healthcare system," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5914–5925, 2021.

[38] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: a state of the art survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 858–880, 2019.

[39] T. Okamoto, "Cryptography based on bilinear maps," in *Proceedings of the International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pp. 35–50, Las Vegas, NV, USA, February 2006.

[40] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 127–144.

[41] A. Boldyreva, C. Gentry, A. O. Neill, and D. H. Yum, "Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 276–285.

[42] J. Shao, Z. F. Cao, K. liang, and H. Lin, "Proxy re-encryption with keyword search," *Information Sciences*, vol. 180, no. 13, pp. 2576–2587, 2010.

[43] J. W. Liu, Z. H. Zhang, X. F. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 332–342, 2014.