IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SECURITY AND PRIVACY IN EMERGING DECENTRALIZED COMMUNICATION ENVIRONMENTS

Received August 9, 2019, accepted September 10, 2019, date of publication September 23, 2019, date of current version October 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943153

Cloud-Assisted EHR Sharing With Security and Privacy Preservation via Consortium Blockchain

YONG WANG[1], AIQING ZHANG[1], (Member, IEEE), PEIYUN ZHANG[2], (Senior Member, IEEE), AND HUAQUN WANG[3], (Member, IEEE)

[1]School of Physics and Electronic Information, Anhui Normal University, Wuhu 241002, China
[2]School of Computer and Information, Anhui Normal University, Wuhu 241002, China
[3]School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Corresponding author: Aiqing Zhang (aqzhang2006@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61601005, Grant 61872006, and Grant 61872192, in part by the Natural Science Foundation of Anhui Province under Grant 1808085MF164, in part by the Anhui Provincial Key Laboratory of Network and Information Security under Grant AHNIS2018003, in part by the Scientific Research Staring Foundation of Anhui Normal University under Grant 2018XJJ40, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20181394.

ABSTRACT The sharing of electronic health records (EHRs) has great positive significance for research of disease and doctors' diagnosis. In recent years, cloud-based electronic medical record sharing scheme has brought a lot of conveniences, but the centralization of cloud exposes threats inevitably to data security and privacy preservation. Blockchain technology can be seen as a promising solution to address these problems on account of its unique propertis of decentration, anonymity, unforgeability and verifiability. In this paper, we propose a blockchain based secure and privacy-preserving EHR sharing protocol. Data requester can search desired keyword from data provider to find relevant EHRs on the EHR consortium blockchain and get the re-encryption ciphertext from cloud server after getting the data owner's authorization. The scheme mainly uses searchable encryption and conditional proxy re-encryption to realize data security, privacy preservation, and access control. Furthermore, proof of authorization is designed as the consensus mechanism for consortium blockchain to guarantee system's availability. Security analysis demonstrates that the proposed protocol can achieve security goals. Besides, we emulate the cryptographic primitives and implement the proposed scheme on Ethereum platform. Performance evaluation shows that the proposed scheme has high computational efficiency.

INDEX TERMS Electronic health records, data sharing, blockchain, data security, privacy preservation.

I. INTRODUCTION

With high-speed development of information technology and Internet technology, Electronic Health Records (EHRs), as a replacement of traditional manuscript patient's health records on paper, solve the problems of paper that easy to lose, difficult to save for a long time and not easy to carry. For the research of disease, doctors or medical institutions need abundant EHRs which contain similar or related disease to compare and analyze for seeking better therapeutic methods [1]. For a patient, he/she may not be able to remember his/her medical history or can't describe detailed symptoms. EHR sharing is a promising solution for these problems, which can help doctors know more about patients, such that improving the accuracy of disease diagnosis.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

EHR sharing has attracted extensive attentions and researches from industry and academia, where the most noteworthy issues are privacy preservation, data security and interoperability [2]. First, EHRs include personal and high privacy-sensitive information, thus privacy preservation is the guard of patients' reputation and benefit. Second, only the authentic data in EHRs can reflect the real situation and promote the development of medical treatment. On the contrary, the forged or modified data reduces the effective utilization of EHRs. Additionally, the interoperability can help patients to control the access right of their EHRs and enhance mobility of EHRs between different healthcare institutions.

In response to these questions, cloud technology has been put forward for health data storage, management and sharing [3]–[8]. These works use different cryptographic algorithms and cloud technology to design access control schemes

for EHR sharing to realizing privacy preservation and data security. Although these works provide promising solutions for EHR sharing in cloud environment and pay high attention to data security and privacy protection, there still remains one severe challenge: the cloud is supposed to be trusted in storing and managing the data. The pattern of cloud-based EHR sharing relies on third-party which may steal, leak, tamper or abuse the data once they are under attacks or lack of monitoring. Despite that many cryptographic primitives are applied in different schemes [4]–[7], the problem of single point failure can't be solved due to the centralization characteristic of cloud.

Fortunately, blockchain technology as a distributed public ledger is a prospective solution to figure out security issues in EHR sharing after the cloud-based system [9]. Due to the fact that the blockchain is open and transparent, EHR sharing based on blockchain can help patients to control access permission and supervise the utilization of their EHRs. Even though blockchain technology has a series of advantages for building EHR sharing system, we still face the following challenges: 1) How to achieve data privacy preservation with EHR searchability in blockchain? 2) How to realize that only the patient and authorized entities can access the EHR? 3) How to design the data structure and consensus mechanism of consortium blockchain established by different entities to maintain the system running efficiently and normally?

In order to address the above challenges, we propose a cloud-assisted blockchain scheme which combines searchable encryption and proxy re-encryption technology to realize privacy preservation and data security for EHR sharing. In this work, the keyword ciphertext stored in consortium blockchain ensures users to find expected EHRs and protects data security with searchability. Besides, the combination with proxy re-encryption and cloud technology is adopted to guarantee that only authorized entities can access the EHRs. We also design a suitable data structure and consensus mechanism of consortium blockchain to ensure high-efficiency, reliability, and safety of the entire system. In summary, the contributions of our scheme are threefolds as follows.

- We propose a new framework for cloud-assisted EHR storage and sharing with privacy preservation and data security based on consortium blockchain. The cloud is used to store patients' EHR ciphertext while the consortium blockchain keeps reocrds of keyword ciphertext for data searching and sharing.
- We design the following core components for consortium blockchain: network model, data construction, and consensus mechanism. We define different entities, and stipulate their authority according to the demand of our system in the network. We design the block structure and transaction structure and incorporate cryptography primitives to store data securely. Furthermore, we put forward proof of authorization as the consensus mechanism for consortium blockchain.
- We present a cloud-assisted secure and privacy-preserving EHR sharing protocol based on consortium blockchain. Only the authorized data requesters who have searching trapdoor are allowed to acquire the keywords and related information. Moreover, the authorization and other access services are accomplished by the blockchain accounts, which ensures identity privacy protection. Also, the cloud re-encrypts the EHR ciphertext and sends the re-encrypted ciphertext to specified data requester when they come to an agreement with the patient.

The structure of the paper is organized as follows. An overview on existing works related to our research is presented in section II. Section III gives the key technologies prepared for our scheme. Section IV constructs the system architecture, EHR consortium blockchain and analyzes the threat model and security goals. The data structure and consensus mechanism of EHR consortium blockchain are designed in section V. Section VI describes details of the protocol and security proof. Later, we discuss how the protocol achieves security goals in section VII. Furthermore, we compute the computational overhead and communication overhead and evaluate the performance of our system by implementing it on Ethereum platform in section VIII. Finally, section IX summarizes the paper and looks ahead to the future.

II. RELATED WORK

In this section, we discuss works that focus on EHR sharing with the help of cloud technology and blockchain technology.

A. EHR SHARING WITH CLOUD

In order to achieve data security during the process of EHR sharing, some access control schemes based on cloud were introduced in [3]–[5]. A new method of fine-grained access control called ciphertext-policy attribute-based signcryption and secure sharing of personal health records in cloud computing was proposed in [3]; In [4], an efficient and secure fine-grained access control scheme was presented which can realize authorized users to access EHRs in cloud storage. It supports some specific physicians to write on EHRs; [5] proposed a hierarchical comparison-based encryption scheme and developed a dynamic policy updating scheme by using the proxy re-encryption technique to achieve dynamic access control in cloud-based EHR systems.

For improving the searchability and interoperability of EHR sharing, [6] proposed a new cloud-based EHR system supporting fuzzy keyword search for secure data sharing and effective utilization of the EHRs; [7] utilized conjunctive keyword search with proxy re-encryption to build a secure EHR searching scheme for data sharing between different medical institutions. Moreover, [8] proposed a general framework for secure sharing of EHRs that patients are allowed to securely store and share their EHR in the cloud server and doctors can access the EHRs in cloud.

B. EHR SHARING WITH BLOCKCHAIN

With the development of blockchain technology, its decentralized, traceability and anonymous characteristics have been widely concerned in applications of medical industry issues. At present, many scholars are focusing on the privacy and security in EHR sharing based on blockchain technology.

In order to help patients use and share their personal health data conveniently and safely, Amofa *et al*. [10] presented a blockchain architecture to realize the security control of personal data in health information exchange by matching intelligent contracts with user-generated acceptable policies. The architecture minimized data security risks by designing a mechanism to control the shared data. X. Zheng *et al*. [11] proposed a conceptual design for personal continuous-dynamic health data sharing based on blockchain technology. It is supplemented by cloud storage, so as to share information related to personal health in a safe and transparent way. In [12], an identity and access management system using blockchain technology to support the authentication and authorization of entities in digital systems was proposed. This system described the application of blockchain in Hyperledger Fabric framework for identity authentication and access management. Moreover, Guo *et al*. [14] proposed an attribute-based signature scheme with multiple authorities to ensure the effectiveness of encapsulated EHRs in the blockchain. In this scheme, the patient endorsed the message according to the attributes and only provided the evidence that he had attested to it.

Some schemes combine cloud technology with blockchain technology to improve the security of EHR sharing. Cao *et al*. [13] proposed a cloud-assisted secure eHealth system, using blockchain technology to protect outsourced EHRs in cloud from illegal modification. The key idea of this system was that EHRs can only be outsourced by authenticated participants. Each operation on the outsourced EHRs was integrated into public blockchain as a transaction. Liu *et al*. [18] proposed a blockchain-based privacy-preserving data sharing scheme, namely, BPDS. In BPDS, the cloud was used to store the original EMRs securely and a tamper-proof consortium blockchain was designed to share the EMR indexes. The scheme used this way to reduce the risk of medical data leakage. The use of consortium blockchian ensures that the EMRs cannot be modified discretionarily. In [19], a storage scheme and service framework were proposed for storing, sharing and using medical data based on blockchain and cloud. In this scheme, blockchain-based personal medical data applications can provide a patient medical information service without violating privacy concerns.

Another line of work focused on handling the privacy and access control of EHR sharing on blockchain. Reference [15] proposed a confidential data sharing model to support personal health record system based on blockchain technology and proxy re-encryption method. The model solved three important problems: privacy of on-chain data, limited storage for large medical data and consent revocation. Reference [16] presented a blockchain-based system architecture to achieve an auditable medical data sharing and healthcare data access permission handling. In other aspects, Chen *et al*. [17] proposed a blockchain-based searchable encryption scheme for electronic medical record sharing to improve data searchability. In this scenario, the construction of EHR indexes stored in the blockchain were complex logical expressions, so that data users can use those logical expressions to search the indexes. Taking advantage of the decentralized property of blockchain, data owners had complete control over who can see their EHRs. The blockchain technology guarantees data integrity, anti-interference, and traceability.

Different from the above works, Zhang and Lin *et al*. [29] proposed a multi-typed blockchain-based secure and privacy-preserving PHI sharing (BSPP) for diagnosis improvements. In BSPP, the private blockchain was used to store PHI for hospital and the consortium blockchain was responsible for recording the secure indexes of the PHI. The scheme used public key encryption with keyword search for realizing data security and privacy preservation of data sharing on consortium blockchain.

The aboving works proposed various EHR sharing schemes from different aspects. Generally, they presented an idea or concept while without detail solutions for a specific application scenarios. In our work, we combine keyword searchable encryption and proxy re-encryption technology to realize privacy-preserving and secure data sharing for EHR sharing based on consortium blockchain technology and cloud storage. Furthermore, we design the protocol in details.

III. PRELIMINARIES

In this section, we give the technical preliminaries required in this paper.

A. BILINEAR MAPS

Let G_1 and G_2 be two cyclic groups of the same prime order q. A bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear map if it satisfies the following properties:

1) $\hat{e}(aR, bS) = \hat{e}(R, S)^{ab}$, for all $R, S \in G_1$ and $a, b \in Z_q^*$.
2) $\hat{e}(R, S) = \hat{e}(S, R)$.
3) $\hat{e}(R + S, T) = \hat{e}(R, T)\hat{e}(S, T)$, for all $R, S, T \in G_1$.
4) $\exists R, S \in G_1, \hat{e}(R, S) \neq 1_{G_2}$.
5) $\hat{e}$ can be efficiently computed.

B. COMPLEXITY ASSUMPTIONS

Definition 1: Elliptic Curve Discrete Logarithm Problem (ECDLP). We suppose that E is an elliptic curve. The primitive element is P and X is another element in elliptic curve. Given ♯E as the number of points on the curve, the ECDLP is looking for the integer b, where $1 \leq b \leq \sharp E$, *which satisfies the following:*

$$\underbrace{P + P + \cdots + P}_{b} = bP = X$$

In cryptosystems, the private key is usually an integer b and the public key X is a point on the curve with coordinates $X = (x_X, y_X)$.

ECDLP Assumption. It is assumed that it is difficult to solve the ECDLP in polynomial time.

Definition 2: Decision Linear Diffie-Hellman Problem (DLDH). We denote an elliptic curve E and consider a cycle group G_1 of prime order q. Let P_1, P_2, P_3 be random elements in G_1 and a_1, a_2, a_3 random numbers in Z_q^*. The DLDH problem is defined as follows: Given a tuple $(P_1, P_2, P_3, a_1P_1, a_2P_2, a_3P_3) \in G_1$ as input, output 1 if $a_3 = a_1 + a_2$ and 0 otherwise. We define the advantage of an algorithm $\mathcal{A}$ to deciding the DLDH problem in G_1 as:

$$Adv_1 = \left| \begin{array}{l} \Pr[A(P_1, P_2, P_3, a_1P_1, a_2P_2, (a_1 + a_2)P_3) = 1] \\ - \\ \Pr[A(P_1, P_2, P_3, a_1P_1, a_2P_2, a_3P_3) = 1] \end{array} \right|$$

DLDH Assumption. If the probability of any t-time adversary successfully solving the problem is $Adv_1 \leq \varepsilon$, where ε is negligible, it is assumed that it is hard to solve the DLDH problem in polynomial time.

Definition 3: Modified Decisional Bilinear Diffie-Hellman Problem (m-DBDH). We denote E an elliptic curve and the primitive element is P. Consider cycle group G_1 and G_2 of prime order q. The m-DBDH is defined as follows: Given a tuple $(P, cP, dP, T) \in G_1^3 \times G_2$ as input, where $c, d \in Z_q^*$, decide whether $T = \hat{e}(P, P)^{d/c}$.We define the advantage of an algorithm $\mathcal{A}$ to deciding the m-DBDH problem as:

$$Adv_1' = \left| \begin{array}{l} \Pr[A(P, cP, dP, e(P, P)^{d/c}) = 1] \\ - \Pr[A(P, cP, dP, T) = 1] \end{array} \right|$$

m-DBDH Assumption: If the probability of any t-time adversary successfully solving the problem is $Adv_1' \leq \varepsilon$, where ε is negligible, it is assumed that it is difficult to decide the m-DBDH in probabilistic polynomial time.

C. PUBLIC KEY ENCRYPTION WITH CONJUNCTIVE KEYWORD SEARCH

The public key encryption with conjunctive keyword search enables data requesters to search a document containing several keywords over a public key encryption setting. The scheme is defined as following algorithms [20].

- $KeyGen(1^k)$: Given a security parameter 1^k as input, it outputs public/private key pair (pk, sk).
- $PECK(pk, W)$: It selects a keyword set $W = \{w_1, w_2 \cdots, w_n\}$. It uses the public key to produce a searchable keyword encryption C_w for W.
- $Trapdoor(sk, Q)$: It takes the receiver's private key sk and the keyword query $Q = (\Omega_1, \Omega_2, \cdots, \Omega_t)$ as input, and computes the trapdoor T_Q for the conjunctive search of a given keyword query.
- $Test(pk, C_w, T_Q)$: It takes as input the public key pk, searchable keyword encryption C_w and the trapdoor T_Q. If Q is included in C_w, the server outputs "yes", otherwise "no".

D. CONDITIONAL PROXY RE-ENCRYPTION

Conditional proxy re-encryption is a scheme which only allows the proxy with a re-encryption key to convert ciphertext satisfying a concrete condition. The re-encryption ciphertext encrypted by a delegator's public key and condition c can be decrypted by the delegatee who satisfies the condition c with his/her private key. The scheme consists of the following algorithms [21].

- $Setup(k)$: Given a security parameter k as input, the algorithm outputs the system's public parameter.
- $KeyGen(i)$: This algorithm generates a public-private key pair (pk_i, sk_i) for user.
- $Enc(sk_s, pk_i, m)$: It takes the sender s's private key, the receiver i's public key and plaintext m as input, and returns ciphertext C_m.
- $ReKeyGen(pk_s, sk_i, pk_j)$: The delegator i generates a re-encryption key by using his/her private key, the sender s's public key and delegatee j's public key.
- $ReEnc(C_m, rk)$: This algorithm takes as input ciphertext C_m and re-encryption key rk, and outputs the re-encryption ciphertext C_m'.
- $Dec(C_m', sk_j)$: It takes the re-encryption ciphertext C_m' and delegatee j's private key as input, and returns the plaintext m.

E. BLOCKCHAIN TECHNOLOGY

Blockchain is an ordered list of records linked together through a chain on blocks [22]. It is essentially a decentralized database, which is a new application mode of distributed data storage, point-to-point transmission, consensus mechanism, encryption algorithm, and other computer technologies. It is also a distributed ledger that cannot be tampered or forged by using the cryptography method.

Current blockchain systems can be categorized into three types: Public blockchain, private blockchain, and consortium blockchian [24]. Public blockchain is permissionless blockchain where all records are visible to the public and anyone can take part in the system and access information, for example, Bitcoin, Ethereum. A private blockchain is regarded as a centralized network since an organization fully controls the system. Consortium blockchain is a partially decentralized system since it is managed by several organizations. In consortium blockchain, only those nodes that come from authorized organizations can access data in blockchain. In our work, we conduct EHR data sharing on consortium blockchain. Several hospitals constitute an alliance and create a consortium blockchain, which keeps records of secure indexes for patient's EHR.

In blockchain, the way to reach consensus among untrust-worthy nodes in distributed environment is called consensus mechanism. The consensus mechanism is the core of blockchain technology. Proof of work, proof of stake, practical byzantine fault tolerance and some other consensus mechanism have been proposed for blockchain [24]–[28].

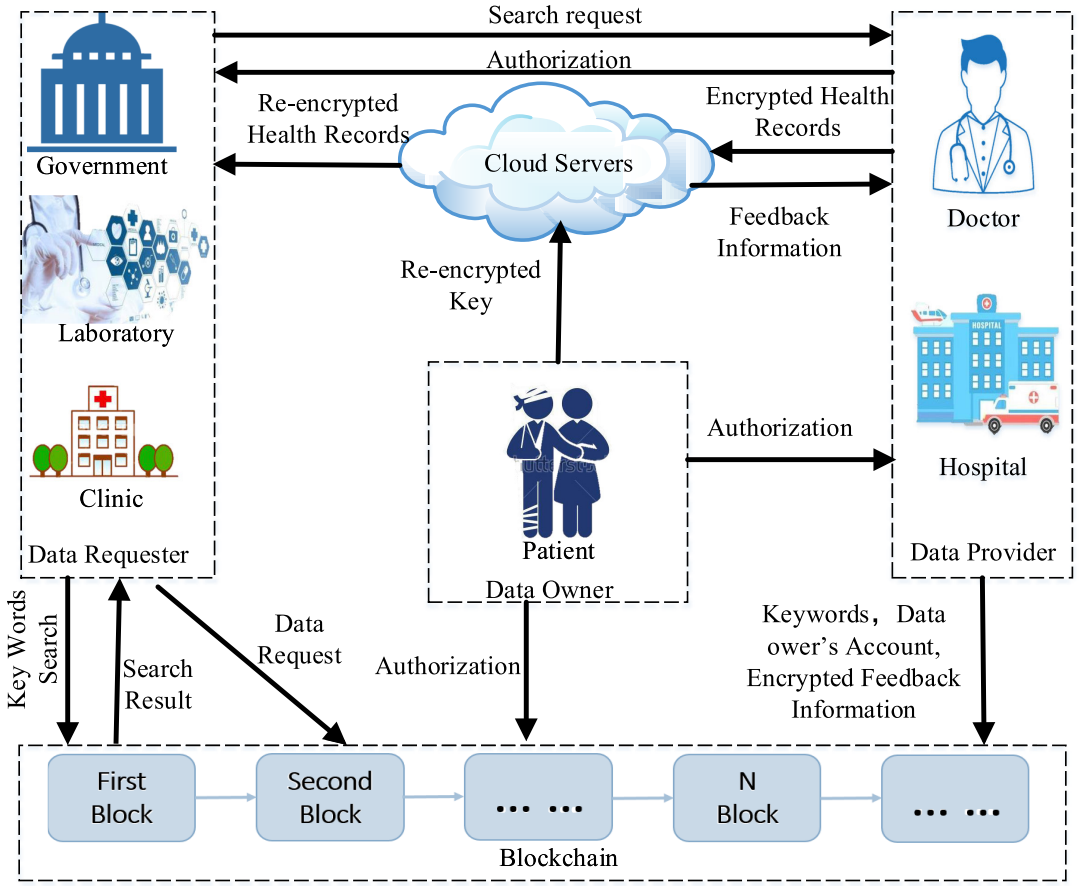

FIGURE 1. System model.

IV. SYSTEM MODEL

In this section, we present the architecture for cloud-assisted consortium blockchain for EHR storing and sharing system. And then, we analyze the threats and put forward our security goals.

A. SYSTEM ARCHITECTURE

There are five entities in the proposed framework: Data owners (DO), data providers (DP), cloud server (CS), blockchain (BC), data requesters (DR), as shown in Fig. 1.

1) DATA OWNERS

Date owners refer to patients who visit doctors in hospitals or medical institutions for medical service. The electronic health records including data of individual privacy will be produced after their interactions. As the source of health record, DO has the ownership and control rights for the data. They must register an account for data sharing on EHR consortium blockchain. The DP can upload health record to cloud after getting DO's authorization. Data requesters need DO's permission for accessing the data.

2) DATA PROVIDERS

Data providers are doctors or administrators of hospitals who manager EHRs. When receiving a patient's authorization, they encrypt the health record and upload files to cloud server. Afterwards, they conduct a data transaction consisting of keyword ciphertext for EHR and DO's account address and send it to the transaction pool. They act as data transaction senders in blockchain, as shown in Fig. 2. If a new DP wants to join the blockchain, he/she has to take three steps:

- Register an account in EHR consortium blockchain.
- Submit a recommendation letter signed by one commissioner and send it to all of the commissioners.
- Get at least 2/3 of the authorizations from commissioners.

3) CLOUD SERVER

Cloud server is in charge of storing encrypted EHR provided by DP. It is also responsible for sending the file location to DO's account in EHR consortium blockchain. It is honest but curious about the data. In addition, it takes responsibility for re-encrypting EHR using re-encryption key.

4) DATA REQUESTERS

Data requesters refer to government, laboratory, clinic, and so on, who need to access patient's EHR. They have to get search trapdoor from DP and search for keywords in the blockchain at first, and then send a request to DO after getting

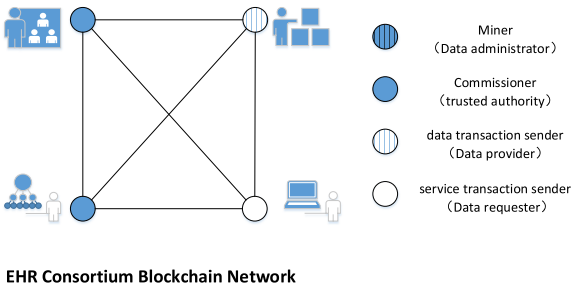

FIGURE 2. EHR consortium blockchain network.

search result. Once they get DO's authorization, they will receive the re-encrypted health record from cloud server. Their operation will generate service transactions that will be put into transaction pool, thus they act as service transaction senders in blockchain, as shown in Fig. 2. They can join or exit blockchain network anytime as the ordinary users. They can see the whole consensus process and enjoy the services of the system.

B. CONSTRUCTION OF EHR CONSORTIUM BLOCKCHAIN

The proposed EHR consortium blockchain is composed by blocks which include keyword ciphertext, DO's account address, DP's signatures, and so on. In the blockchain, different members have different access right. Data requesters can perform keyword search and send data access request transactions to blockchain for data sharing. In blockchain network, the nodes should achieve a consensus to generate new blocks. Patients' information is in ciphertext and unlinked to their identities, hence the blockchain can protect their privacy effectively.

The EHR consortium blockchain is composed by four different nodes: commissioner (trusted authority), miner (data administrator), data transaction sender (data provider), service transaction sender (data requester), as show in Fig. 2.

1) COMMISSIONERS

Several hospitals, clinics and medical center constitute an alliance committee and create a EHR consortium blockchain. Each organization owns a commissioner as the member of the alliance committee to execute their decisions. The commissioner is responsible for recommending and approving new data administrator, data provider and verifying valid transactions and blocks. Each commissioner have equal status in whole network. In practice, the commissioner can act as data administrator or data provider. Every block is sent to all of the commissioners for verification after at least 2/3 of the authorizations are received, the block will be marked as valid block.

2) DATA ADMINISTRATORS

Data administrators are generated by random selecting from commissioners as a miner in the blockchain. They take charge of packing transactions and producing blocks. Each cooperative organization must provide at least one data administrator candidate for maintaining normal operation of blockchain. Once getting the appointment, they will gather data transaction and service transaction from transaction pool and pack them into a block. Then, they sign the block and send them to all of the commissioners. When a valid block is added to blockchain network, they will get the deserved reward.

3) DATA TRANSACTION SENDERS

Data providers undertake the responsibility of data transaction sender. They were introduced in system architecture.

4) SERVICE TRANSACTION SENDERS

Data requesters undertake the responsibility of service transaction sender. They were introduced in system architecture.

C. THREAT MODEL AND SECURITY GOALS

In our scheme, cloud servers are semi-trusted. It is honest but curious about electronic health record. They may try to decrypt the ciphertext. Some malicious opponent may intercept, modify or counterfeit the health records during the transmissions. The cloud and data requesters may collude to deduce the plaintext of EHR.

Considering the above threat model, security goals are as follows:

1) DATA CONFIDENTIALITY AND INTEGRITY

The patient's health records can't be read or modified by other entities without data owner's authentication, whatever it is stored in cloud server or transmitted in the public channel.

2) ACCESS CONTROL

The data owners have the ability to control the data access. Only getting the data owner's authorization can other entities access the health records.

3) AUTHENTICATION

Data owners should be able to authenticate data providers to ensure that health records come from reliable resource. Data requesters could be authenticated to guarantee legitimate use of data. The cloud server should be able to authenticate data owner, data provider, and data requester.

4) SECURE SEARCH

Data requesters need to get DP's authentication to search interested content in the EHR consortium blockchain. The same keyword in different searching is unlinkability such that the eavesdroppers can't speculate whether two or more EHRs come from the same source.

5) PRIVACY PRESERVATION

Data owner's identity information can't be revealed with EHR and account address. Moreover, the original EHR can't be revealed to illicit entities.

6) COLLUSION RESISTANCE

Even if an entity colludes with the cloud server, they can't access the original EHR without access permission. Besides, the DO and CS can't collude to decrypt the EHR.

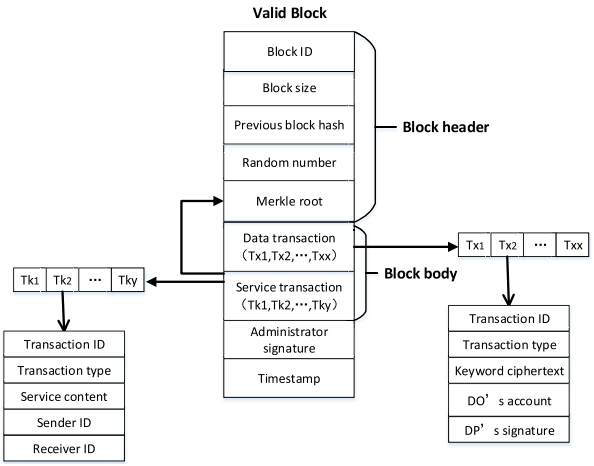

FIGURE 3. Data structure.

Moreover, any two data requesters can not speculate the information of EHR combined with the search trapdoor.

V. EHR CONSORTIUM BLOCKCHAIN DESIGN

A. DATA STRUCTURE

1) BLOCK STRUCTURE

In our scheme, a valid block is composed of block header, block body, data administrator's signature, and timestamp, as show in Fig. 3. Block header contains five components: Block ID, block size, previous block hash, random number, and merkle root. Block ID is used for tracking software or protocol updating which is unique for each block; block size shows how much storage space the block takes up; previous block hash is used to link previous block for avoiding modification; random number is used for appointing the next miner; merkle root is a digital fingerprinting of the transactions set from the block body [23]. Block body has two parts: x data transactions and y service transactions (The optimal design of this quantity is beyond the scope of this article). Data transaction is made up of encrypted EHR and relevant information generated by authorized data provider; service transactions include keyword search, access request, and authorization etc. data administrator's signature helps to track the generator of the block. Timestamp indicates the generation time of the block.

2) TRANSACTION STRUCTURE

Data transaction is made up of transaction ID, transaction type, keyword ciphertext, DO's account, and DP's signature as show in Fig. 3. Transaction ID can help to track source of the transaction; transaction type distinguishes different transactions to guarantee efficient operations; keyword ciphertext is provided for data searching; access request is sent to the DO's account for getting the access authorization; DP's signature provides proof of transaction's validity. A valid data transaction are required by all of the above information.

Service transaction consists of transaction ID, transaction type, service content, sender ID, and receiver ID as show in Fig. 3. Service content may vary from keyword search, exchanging some information between two accounts, sending access request to one's account, and so on. In particular, a valid service transaction must have legal sender and valid receiver. This measure helps to reduce junk information in transaction pool and keep the network running normally and efficiently.

B. CONSENSUS MECHANISM (PROOF-OF-AUTHORIZATION)

We propose a consensus mechanism, named proof of authorization, to build the regulation for consortium blockchain and ensure high-efficiency, reliability and safety of the blockchain network as shown in Fig. 4.

Assume that the number of commissioners is N_c. We assign a random number $M \in [0, N_c - 1]$ to each commissioner in system setup. The system generates a random number M', $0 \leq M' < N_c$, appoints the matched commissioner as data administrator, and produces block in this round. The network will inspect the number of commissioners at next round of consensus and redistribute the number to them.

When a data provider sends data to the EHR consortium blockchain, the data transaction will be stored in the transaction pool at first. In the same way, when data requester submits a request, service transaction will be put into the transaction pool. The appointed data administrator packages x data transactions and y service transactions into a block. Then the block is sent to all of the commissioners.

If a commissioner verifies the block's validity and agrees to authorize the block, he/she will sign the block and return the signature to the data administrator. After receiving at least $2/3N_c$ signatures, data administrator signs on the block and sends it to the NTP server. The NTP server provides the current timestamp, signs and encrypts the new block, then returns the timestamp and signature to the data administrator.

At last, the data administrator generates another random number $M' \in [0, N_c - 1]$ that determines who will be the next data administrator for producing new block and broadcasts to other nodes which can verify the time information of the block. If the total time of the process is less than specified time $T_{\max}$, the block is finally valid. Otherwise, the permissions of producing this block will be turned over to the data administrator $M' + 1 (0 \leq M' < N_c - 1)$. When a valid block is generated, it means that a round of consensus is finished.

VI. PROPOSED PROTOCOL

In this section, we first present an overview of the proposed protocol for cloud assisted EHR sharing with security and privacy-preservation based on EHR consortium blockchain. After that, we describe the proposed protocol in details and security proof.

A. OVERVIEW

The process of the proposed protocol is represented in Fig. 5. The protocol is made up of three layers: Data generation layer, data storage layer, and data sharing layer.

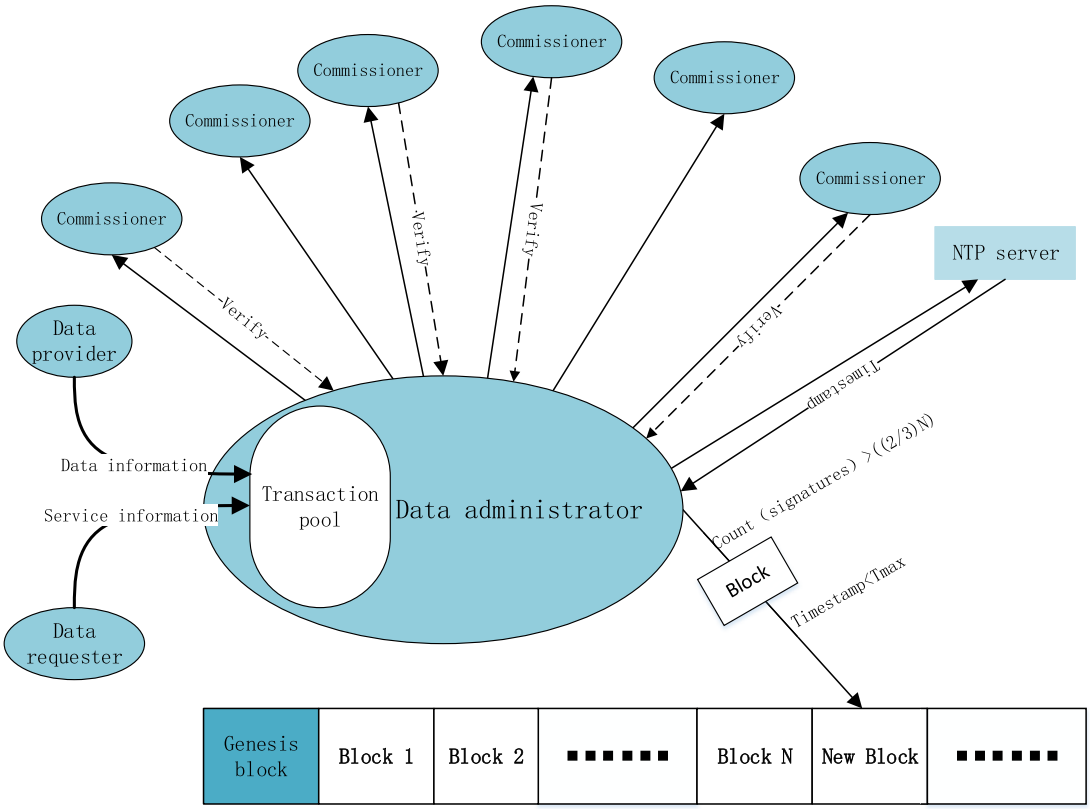

FIGURE 4. Consensus process.

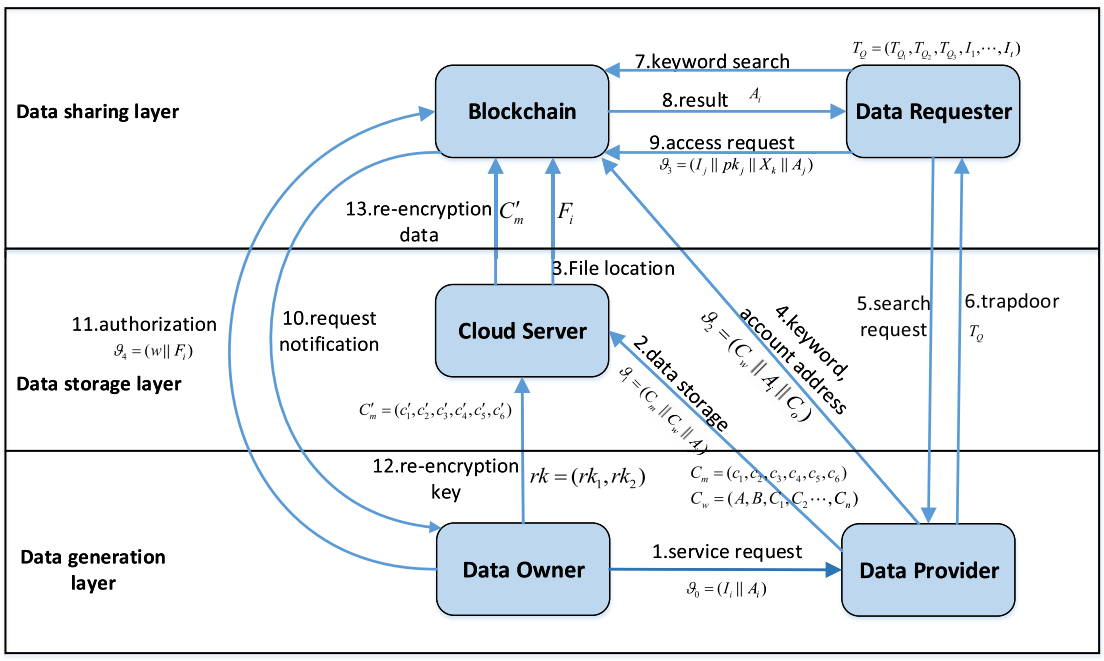

FIGURE 5. Protocol.

When a patient (DO) i with identity I_i arrives at a hospital for a medical service, he/she needs to register an account in the EHR consortium blockchain. An account address A_i and private key generated by the EHR consortium blockchain will be sent to the patient. The patient i sents data packet $\vartheta_0 = (I_i \parallel A_i)$ to a doctor k. The original EHR m for patient

TABLE 1. Algorithm 1 : Data Generation DataGen(m, x_k, pk_i, w_i).

Input:	The original EHR m, DP's private key x_k, DO's public key pk_i, keyword w_i
Output:	EHR ciphertext C_m
1:	selects random number $r \in Z_q^*$
2:	computes $c_1 = r(\sum_{i=1}^{t} H_3(w_i))P$, $c_2 = m\hat{h}^{r/x_k}$, $c_3 = \frac{r}{x_k} pk_i$ $c_4 = r(H_3(m)P_1 + P_2)$, $c_5 = r(H_3(c_1)P_1 + H_3(c_1 \parallel c_2 \parallel c_4)P_2 + P_3)$, $c_6 = rP$
3:	outputs $C_m = (c_1, c_2, c_3, c_4, c_5, c_6)$

TABLE 2. Algorithm 2 : Keyword Index Generation KeyInGen($W = \{w_1, w_2, \cdots, w_n\}$, *params*).

Input:	The keyword set $W = \{w_1, w_2, \cdots, w_n\}$
Output:	Keyword ciphertext C_w
1:	picks random number $u, v \in Z_q^*$
2:	computes $A = vP$, $B = uX_k = ux_kP$, $C_i = vh_i + uf_i$ where $h_i = H_1(w_i)$, $f_i = H_2(w_i \parallel A \parallel B)$ for $1 \le i \le n$
3:	outputs $C_w = (A, B, C_1, C_2 \cdots, C_n)$

i will be generated after interacting with the doctor (DP) k. The DP extracts a series of keyword w_i from the EHR. Then, the DP encrypts m with the patient's public key pk_i, the DP's private key x_k and keyword w_i, getting the EHR ciphertext C_m. In addition, it encrypts w_i with the DP's public key X_k producing keyword ciphertext C_w. After that, the DP sends data packet $\vartheta_1 = (C_m \parallel C_w \parallel A_i)$ to cloud server. The file location F_i will be sent to the DO's account when the cloud server finished storing the data safely. Meanwhile, the DP sends data packet $\vartheta_2 = (C_w \parallel A_i \parallel C_k)$ to the EHR consortium blockchain, where C_k is DP's signature for proof of conformance. Also, DP uses keywords w and his/her private key x_k to produce a trapdoor T_Q for keyword search.

If government, laboratory or clinic (DR) would like to search for some EHR, they first submit a search request to the DP. If the request is allowed, they will get a trapdoor T_Q. Then the DR can find out the matched EHR and obtain the DO's account address A_i by searching on the blockchain with T_Q. Afterwards, they can send data packet $\vartheta_3 = (I_j \parallel pk_j \parallel X_k \parallel A_j)$ to the DO's account for access request. When the DO receives data request notification, they will send an authorization including file location F_i and keyword w_i to DR's account. Additionally, it generates a re-encryption key rk and transmits it to CS, who carries out proxy re-encryption for the required ciphertext. Finally, the DR uses his/her private key sk_j to decrypt the re-encrypted ciphertext C'_m.

B. PROTOCOL DESCRIPTION

The proposed protocol is composed of three phases: System setup and registration, data storage and index generation, data sharing.

Phase 1: System Setup and Registration

System Parameter Generation: Given a security parameter k, the DP generates a prime q and selects a bilinear pairing $\hat{e} : G_1 \times G_1 \to G_2$, where G_1 is an additive cycle group and G_2 is a multiplicative cycle group with the same prime order q. P is the generator of G_1. The DP chooses three different one-way collision-resistant hash function: $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \times G_1 \times G_1 \to G_1$, $H_3 : \{0,1\}^* \to Z_q^*$. Furthermore, the DP randomly selects three parameter $g_1, g_2, g_3 \in Z_q^*$, and computes $P_1 = g_1 P$, $P_2 = g_2 P$, $P_3 = g_3 P$ as the secret key of the system. Additionally, the DP randomly chooses $x_k \in Z_q^*$ as the master private key and computes the public key $X_k = x_k P$. Set $\hat{h} = \hat{e}(P, P)$. Finally, the system parameters is published as $params = (q, P, \hat{e}, G_1, G_2, H_1, H_2, H_3, X_k, \hat{h})$.

TABLE 3. Algorithm 3 : Trapdoor Generation TrapdoorGen($\Omega = (\Omega_1, \Omega_2, \cdots, \Omega_t)$, x_k).

Input:	The keyword set $\Omega = (\Omega_1, \Omega_2, \cdots, \Omega_t)$
Output:	Trapdoor T_Q
1:	randomly chooses $m \in Z_q^*$
2:	computes $T_{Q_1} = mP$, $T_{Q_2} = \sum_{i=1}^{t} mh_i$ and $T_{Q_3} = \frac{m}{x_k} \sum_{i=1}^{t} f_i$, where the corresponding searching keyword set $Q = (\Omega_1, \cdots, \Omega_t)$, Ω_i is the keywords, $h_i = H_1(\Omega_i)$, $f_i = H_2(\Omega_i \parallel A \parallel B)$.
3:	outputs $T_Q = (T_{Q_1}, T_{Q_2}, T_{Q_3})$

Registration: Both the DO and the DR register to DP for joining the system. The DO randomly chooses number $x_i \in Z_q^*$ as his/her private key sk_i and computes $Y_i = x_i P$ as his/her public key pk_i. The DR randomly selects $x_j \in Z_q^*$ as his/her private key sk_j and computes $Y_j = x_j P$ as his/her public key pk_j.

Phase 2: Data Storage and Index Generation

Data Generation: When a DO i visits a hospital and interacts with a DP k, the original EHR $m \in \{0,1\}^*$ will be generated. The DP encrypts m with his/her private key x_k, the DO's public key pk_i and the keyword w_i to generate EHR ciphertext C_m by performing **Algorithm 1**.

Keyword Index Generation: The DP k selects a keyword set $W = \{w_1, w_2, \cdots, w_n\}$ for original EHR and two random values $u, v \in Z_q^*$. It computes the searchable keyword ciphertext $C_w = (A, B, C_1, C_2 \cdots, C_n)$ by performing **Algorithm 2**.

When DP finished data and index generation, the data packet $\vartheta_1 = (C_m \parallel C_w \parallel A_i)$ is stored in cloud server and $\vartheta_2 = (C_w \parallel A_i \parallel C_k)$ is formated as a data transaction.

TABLE 4. **Algorithm 4 : Re-encryption key Generation ReKeyGen(sk_i, X_k, $H_3(\Omega_i)$, pk_j, F_i).**

Input:	The DO's private sk_i, DP's public key X_k, keyword hash $H_3(\Omega_i)$, DR's public key pk_j, file location F_i
Output:	Re-encryption key rk
1:	computes $rk_1 = \frac{1}{sk_i}(\sum_{i=1}^{t} H_3(\Omega_i))X_k = \frac{x_k}{x_i}\sum_{i=1}^{t} H_3(\Omega_i)P$
2:	computes $rk_2 = \frac{1}{sk_i} pk_j H_3(F_i) = \frac{x_j}{x_i} H_3(F_i)P$
3:	outputs $rk = (rk_1, rk_2)$

TABLE 5. **Algorithm 5 : Re-encryption ReEnc(C_m, rk).**

Input:	EHR ciphertext C_m, re-encryption key rk
Output:	re-encrypted ciphertext C'_m
1:	Parses C_m as $(c_1, c_2, c_3, c_4, c_5, c_6)$ and rk as (rk_1, rk_2).
2:	Checks the equality $\hat{e}(c_1, P) = \hat{e}(c_3, rk_1)$.
	If it does not hold, returns $\perp$ which indicates that the ciphertext is not allowed to be re-encrypted.
3:	Checks the equality $\hat{e}(c_6, H_3(c_1)P_1 + H_3(c_1 \parallel c_2 \parallel c_4)P_2 + P_3) = \hat{e}(P, c_5)$.
	If it does not hold, returns $\perp$ which indicates that the ciphertext is not valid.
4:	If both checks above succeed, then cloud server outputs
	$c'_1 = c_1 = r(\sum_{i=1}^{t} H_3(w_i))P, c'_2 = c_2 = m\hat{h}^{\frac{r}{x_k}}$
	$c'_3 = \hat{e}(c_3, rk_2) = \hat{e}(\frac{rx_i}{x_k}P, \frac{x_j}{x_i}H_3(F_i)P) = \hat{h}^{\frac{rx_j H_3(F_i)}{x_k}}$
	$c'_4 = c_4 = r(H_3(m)P_1 + P_2)$
	$c'_5 = c_5 = r(H_3(c_1)P_1 + H_3(c_1 \parallel c_2 \parallel c_4)P_2 + P_3)$
	$c'_6 = c_6 = rP$
5:	outputs $C'_m = (c'_1, c'_2, c'_3, c'_4, c'_5, c'_6)$

Phase 3: Data Sharing

Keyword Search: The DP generates a keyword set $\Omega = (\Omega_1, \Omega_2, \cdots, \Omega_t)$ searching trapdoor T_Q for DR to search desired keyword on the consortium blockchain after receiving the search request from DR. The trapdoor $T_Q = (T_{Q_1}, T_{Q_2}, T_{Q_3})$ is generated by using DP's private key as **Algorithm 3**.

After getting keyword searching trapdoor, the DR searches keyword in the secure indexes on the EHR consortium blockchain to find out the indexes for DO i. The test algorithm is executed on the blockchain by checking the equality $\hat{e}(T_{Q_1}, \sum_{i=1}^{t} C_i) = \hat{e}(A, T_{Q_2}) \cdot \hat{e}(B, T_{Q_3})$. If the equation holds, the blockchain outputs "yes" to the DR and sends DO's account address A_i to him/her. Otherwise, it aborts.

Correctness: We assume that the Ω_i in the keyword trapdoor T_Q and w_i in the ciphertext are equal, the correctness of the test algorithm is verified as:

$$\begin{aligned}\hat{e}(T_{Q_1}, \sum_{i=1}^{t} C_i) &= \hat{e}(mP, \sum_{i=1}^{t}(vh_i + uf_i)) \\ &= \hat{e}(mP, v\sum_{i=1}^{t} h_i + u\sum_{i=1}^{t} f_i) \\ &= \hat{e}(mP, v\sum_{i=1}^{t} h_i) \cdot \hat{e}(mP, u\sum_{i=1}^{t} f_i) \\ &= \hat{e}(vP, m\sum_{i=1}^{t} h_i) \cdot \hat{e}(ux_kP, \frac{m}{x_k}\sum_{i=1}^{t} f_i) \\ &= \hat{e}(A, T_{Q_2}) \cdot \hat{e}(B, T_{Q_3})\end{aligned}$$

Data Access: When the DR gets the DO's account address A_i, he/she will send data packet $\vartheta_3 = (I_j \parallel pk_j \parallel X_k \parallel A_j)$ for access request to the DO's account address A_i. After getting the access request notification, the DO transmits data packet $\vartheta_4 = (H_3(w_i) \parallel F_i)$ for authorization to DR's account address A_j and generates re-encryption key $rk = (rk_1, rk_2)$ for cloud server, where rk_1 and rk_2 are calculated by **Algorithm 4**.

Then the re-encryption key rk is sent to the cloud server to re-encrypt the ciphertext from DP.

Upon receiving the re-encryption key, the cloud server carries out the **Algorithm 5** to generate a re-encrypted ciphertext C'_m.

Then, the cloud server sends the re-encrypted ciphertext $C'_m = (c'_1, c'_2, c'_3, c'_4, c'_5, c'_6)$ to DR's account. The DR j is able to decrypt the re-encrypted ciphertext with his/her private key according to **Algorithm 6**

Correctness: The correctness of the **Algorithm 6** is verified as:

$$\begin{aligned}\tilde{m} &= \frac{c'_2}{(c'_3)^{1/sk_j}} = \frac{m\hat{h}^{\frac{r}{x_k}}}{\hat{h}^{\frac{rx_jH_3(F_i)}{x_k}\cdot\frac{1}{x_jH_3(F_i)}}} = m \\ \hat{e}(c'_4, P) &= \hat{e}(r(H_3(m)P_1 + P_2), P) \\ &= \hat{e}(rP, H_3(m)P_1 + P_2)\end{aligned}$$

$$= \hat{e}(c_6', H_3(m)P_1 + P_2)$$

$$\hat{e}(c_1', P) = \hat{e}(r\sum_{i=1}^{t} H_3(w_i)P, P)$$

$$= \hat{e}(rP, \sum_{i=1}^{t} H_3(w_i)P)$$

$$= (c_6', \sum_{i=1}^{t} H_3(w_i)P)$$

TABLE 6. Algorithm 6 : Decryption $\mathbf{Dec}(C_m', sk_j)$.

Input:	The re-encrypted ciphertext C_m', DR's private key sk_j
Output:	original EHR m or $\perp$
1:	computes $\tilde{m} = \frac{c_2'}{(c_3')^{1/sk_j H_3(F_i)}}$.
2:	checks the equality $\hat{e}(c_4', P) = \hat{e}(c_6', H_3(m)P_1 + P_2)$.
3:	checks the equality $\hat{e}(c_1', P) = \hat{e}(c_6', \sum_{i=1}^{t} H_3(w_i)P)$.
4:	If all of the equation holds, outputs valid message $\tilde{m}$; otherwise outputs $\perp$.

C. SECURITY PROOF

Theorem 1: The proposed PEKS is secure against IND-CR-CKA in the random oracle model assuming the DLDH assumption holds in G_1.

The PEKS includes system setup and registration, Algorithm 2, Algorithm 3. The security model of indistinguishability of ciphertext from random against chosen keyword attacks (IND-CR-CKA) is the same as [20].

Proof: See Appendix A.

Theorem 2: The proposed CPRE is IND-CCA secure in the standard model assuming the m-DBDH assumption is intractable.

The CPRE includes system setup and registration, Algorithm 1, Algorithm 4, Algorithm 5, Algorithm 6. The security model of indistinguishability under Chosen Ciphertext Attacks (IND-CCA) is the same as [21].

Proof: See Appendix B.

VII. SECURITY ANALYSIS

In this section, we demonstrate how the proposed protocol achieves the security goals effectively.

1) THE PROPOSED PROTOCOL CAN ACHIEVE DATA CONFIDENTIALITY AND INTEGRITY

DP encrypts EHR with his/her private key, DO's public key and a series of keywords extracted from the EHR before sending them to cloud server. So the ciphertext can't be decrypted without DP's public key, DO's private key and keywords. The private key is secure under ECDLP assumption. In addition, only the entity that gets DO's authorization is allowed to access the data from cloud server. In practice, DO generates a re-encryption key with DR's public key, keyword, and file location. Then cloud server re-encrypts the EHR ciphertext with re-encryption key. Thus, only the intended DR can decrypt the ciphertext, which enhances data confidentiality. Furthermore, the signatures in each block can achieve data integrity.

2) THE PROPOSED PROTOCOL CAN ACHIEVE ACCESS CONTROL

In our system, DP sends a keyword trapdoor T_Q to authorize DR for keyword searching in EHR consortium blockchain. As $T_Q = (T_{Q_1}, T_{Q_2}, T_{Q_3}) = \frac{m}{x_k}\sum_{i=1}^{t} f_i$, it includes DP's private key. It is used for searching the matched keywords which are encrypted by the DP's public key. Thus, DP can control data search.

Moreover, when DO agrees DR to access his/her data, he/she will send them a packet which contains file location and keyword. Meanwhile, DO generates a re-encryption key rk and sends it to CS for performing proxy re-encryption. Note that in $rk = (rk_1, rk_2)$, $rk_1 = \frac{1}{sk_i}(\sum_{i=1}^{t} H_3(\Omega_i))X_k$, $rk_2 = \frac{1}{sk_i} pk_j H_3(F_i)$, which are related to DO's private key, DP's and DR's public key, file location, and keyword. In this way, only the authorized DR can decrypt the re-encrypted ciphertext. Therefore, DO is able to control the access of his/her data.

3) THE PROPOSED PROTOCOL CAN ACHIEVE AUTHENTICATION

Our scheme can achieve both identity authentication and data authentication. The EHR consortium blockchain network distinguishes different nodes and their legality. DR can affirm whether the ciphertext sent by CS is the expected data by examining whether he/she has the ability of decrypting the ciphertext. The re-encryption key is generated by DO's private key, DP's and DR's public key, file location and keyword. It ensures that only the EHR ciphertext which is stored in designated location and encrypted by DO's public key can be re-encrypted. Only the authorized DR can decrypt the target ciphertext by using his/her private key with right file location and keyword.

4) THE PROPOSED PROTOCOL CAN ACHIEVE SECURE SEARCH

The keywords for searching are encrypted by DP's public key in consortium blockchain. DR has to get a searching trapdoor from DP for searching target keyword. So, during the process of DR searching, other entities can't know the search keywords and the searching result. According to **Theorem 1**, our scheme is IND-CR-CKA secure in random oracle model. The attackers can't find the relationship between encrypted keyword and searching trapdoor even though they get the trapdoor. As our scheme is IND-CCA secure in the standard model, according to **Theorem 2**, the cloud server only executes proxy re-encryption for prescriptive original ciphertext and sends it to specific DR. It is not allowed to obtain any information about original EHRs. Furthermore, DP is only authorized to access keywords without revealing other information.

5) THE PROPOSED PROTOCOL CAN ACHIEVE PRIVACY PRESERVATION

In the process of data transmission, the entity sends and receives data packets via his/her account in blockchain. The blockchain account is anonymous and unlinkable to real identity. So, the anonymity of blockchain can protect the public information from divulging the real identity of entities. Besides, during the process of keyword search, it will not reveal any information about DO. During the process of proxy re-encryption, the CS can't deduce the real identity of DO from the EHR ciphertext and re-encryption key.

6) THE PROPOSED PROTOCOL CAN ACHIEVE COLLUSION RESISTANCE

On the one hand, the EHR ciphertext is encrypted by DP's private key, DO's public key, and keyword. Even though DR colludes with CS, they can't decrypt any information from the ciphertext because they do not have DO's private key. On the other hand, the re-encryption key is generated with DR's and DP's public key, file location and keyword, so the re-encryption ciphertext can't be decrypted without DR's private key. Thus, illegal DR isn't able to collude with CS to access the data.

VIII. IMPLEMENTATION AND PERFORMANCE EVALUATION

In this section, we firstly illustrate the parameters setting and platform setting and compare the security properties of the proposed scheme with other schemes. Then, we analyze the communication overhead of the proposed protocol and compare it with another scheme. Finally, we implement the proposed scheme on Ethereum platform and evaluate its performance.

A. PARAMETERS AND PLATFORM SETTING

The system parameter $k = 128$. We use Type A pairing on the elliptic curve $y^2 = x^3 + x$ over the field F_p for some prime $p = 3 \bmod 4$, the same setting as [29].The cryptographic primitives are implemented using Java language on a computer with Intel(R)Core(TM)i5-6500 CPU @ 3.20GHz 3.19GHz, 4.00 GB RAM, Windows 10 operating system.

We use Ganache(client version) to build a private test blockchain on macOS system. The data is written into smart contracts by using solidity language and uploaded to the Ethereum blockchain. The solidity compiler is solc@ 0.4.25 and the smart contracts test framework is mocha@6.2.0. Since solidity can not output the time cost of publishing smart contracts to blockchain, the Web3js library of Nodejs(Node is a development platform that lets JavaScript run on the server side) is used to interact with smart contracts on the blockchain and test the time cost of sending transactions. The specific configurations are shown in Table 7.

TABLE 7. Configurations of ethereum test blockchain.

Operation system	macOS Mojave version 10.14.5
CPU	Intel(R)Core(TM) i5-5350U CPU @1.80GHz
RAM	8 GB 1600MHz DDR3
Program language	Solidity, Javascript
Solidity compiler	solc@ 0.4.25
Test framework	mocha@6.2.0
Interactive platform	Web3@1.2.0
Ethereum platform	Ganache v2.0.2-beta.0

B. COMPARISONS OF SECURITY PROPERTIES

We compared the security properties of the proposed scheme with cloud-based schemes Liu [3], Wang [7], and blockchain-based schemes Sandro [10], Liu [18]. From the table 8, we can find all the schemes can achieve the properties of access control and privacy preservation, which is crucial security objectives in EHR sharing system.

C. COMMUNICATION OVERHEAD

We donate $|G_1|$, $|G_2|$ the size of an element in group G_1 and G_2, $|Q|$ the size of the elements in Z_q^*, $|\sigma|$ the size of signatures. The size of blockchain account is 32 bytes. The communication overhead is generated during the process of data generation, keyword search, and data access. At the data generation phase, the communication overhead between DP and CS comes from data packet ϑ_1. The packet ϑ_1 is made up of C_m, C_w and A_i, the total length is $(n+4)|G_1| + (n+1)|G_2| + 3|Q| + 32$ bytes. Additionally, the communication overhead between DP,DO, and DR is caused by ϑ_2, which is composed of C_w, A_i and C_k. The length of ϑ_2 is $(n+2)|G_1| + n|G_2| + |\sigma| + 32$ bytes. During the process of keyword search, the communication overhead of DR is $2|G_1| + |G_2|$ bytes. At the data access phase, the communication overhead is $6|G_1| + 3|G_2| + 2|Q| + 64$ bytes, which is caused by ϑ_3, ϑ_4, rk and C'_m, as shown in Table 9.

We compare our communication overhead with Zhang [29]. From Table 9, we can find that our communication costs in the process of data access is higher than in Zhang [29]. Nevertheless, in the process of data generation and keyword search, our communication overhead is lower. This is because we use account on the blockchain in replace of pseudo identity. Moreover, our scheme store EHR ciphertext in cloud that avoids the communication overhead of proof of conformance in the private blockchain.

D. IMPLEMENTATION AND COMPUTATIONAL OVERHEAD

In order to quantify the operation time, we evaluate the performance of cryptographic primitives on the platform shown in section VIII.A. We record the computational overhead of algorithms by setting different keyword amounts in Table 10.

In our protocol, the *system setup* and *registration* phase are simulated by the algorithm ***BuildSystem***. The ***DataGen*** algorithm is used to encrypt original EHRs and generate ciphertext C_m. The ***KeyInGen*** algorithm is responsible for generating searchable keyword ciphertext C_w. The DR gets keyword searching trapdoor T_Q from DP, searches the expected keyword and the matching test is executed in ***KeywordSearch*** algorithm. The re-encryption key rk and

TABLE 8. Comparison of security properties.

Properties	Liu[3]	Wang[7]	Sandro[10]	Liu[18]	Proposed scheme
Blockchain based	×	×	√	√	√
Access control	√	√	√	√	√
Authentication	×	√	×	×	√
Privacy preservation	√	√	√	√	√
Secure search	×	√	×	√	√
Collusion resistance	√	×	×	×	√

TABLE 9. Comparison of the communication overhead among different schemes.

Stage	The proposed	Zhang
Data storage	$(2n+4)\|G_1\|+\|G_2\|+3\|Q\|+32$	$(n+6)\|G_1\|+\|G_2\|+3\|Q\|+59$
Index storage	$(2n+2)\|G_1\|+\|\sigma\|+32$	$8((n+6)\|G_1\|+\|G_2\|+2\|Q\|)+90$
Keyword search	$3\|G_1\|$	$6\|G_1\|+2\|Q\|$
Data access	$6\|G_1\|+2\|G_2\|+3\|Q\|+64$	32

TABLE 10. Computational overhead of cryptographic algorithms(in *ms*).

	Algorithms	$BuildSystem$	$DataGen$	$KeyInGen$	$KeywordSearch$	$ReKeyGen$	$ReEnc$	Dec
n=10	Average time	43	166	112	202	102	40	45
	Max Time	259	299	141	254	141	78	63
	Min Time	31	147	93	184	82	25	31
n=50	Average time	42	577	502	981	495	38	44
	Max Time	251	632	577	1103	578	57	63
	Min Time	26	528	470	923	460	22	28
n=100	Average time	45	1078	1010	2001	1004	39	44
	Max Time	150	1218	1156	2266	1144	64	68
	Min Time	28	1028	936	1873	38	26	27

re-encrypted ciphertext C'_m are generated by algorithms ***ReKeyGen*** and ***ReEnc*** respectively. The ciphertext C'_m is decrypted by ***Dec*** algorithm.

Due to the fact that computational overhead of some algorithms are related to keyword amounts, we implement the algorithms by setting $n = 10$, $n = 50$, and $n = 100$, respectively. From table 10, we can find out that the time cost of ***DataGen***, ***KeyInGen***, ***KeywordSearch*** and ***ReKeyGen*** algorithms increase with the size of keyword amounts. Because these algorithms contain keyword information and carry out some calculation about hash function of keyword. However, the ***BuildSystem***, ***ReEnc*** and ***Dec*** algorithms are not affected by keyword set.

The length of data package is a critical factor affecting the time cost of sending a transaction in the blockchain. According to section VIII.C, the length of data package ϑ_2 in keyword index generation phase is $(n+2)\,|G_1|+n\,|G_2|+|\sigma|+32$ and ϑ_3, ϑ_4 in data sharing phase is $6\,|G_1|+3\,|G_2|+2\,|Q|+64$. The $|G_1|$, $|G_2|$, $|\sigma|$, $|Q|$ are 64 bytes, 384 bytes, 32 bytes, and 32 bytes respectively. Thus, the size of transactions $Tx_1 = 448n + 192$ bytes and $Tx_2 = 1664$ bytes. As the Tx_1 is related to keyword amounts n, we implement the transactions on Ethereum platform by setting $n = 10$, $n = 50$, and $n = 100$. The time cost are shown in table 11.

From table 11, we can know that the time cost of sending transactions to the blockchain is proportional to the length of data package. So, the amounts of the keyword set should not be too large to improve the efficiency of the transactions. Furthermore, the gas consumption increases with the increase of the length of data package. But, the consumption of gas is small and acceptable.

TABLE 11. Time cost and gasUsed of transactions.

$Transtions$	Tx_1			Tx_2
	$n=10$	$n=50$	$n=100$	
Length(bytes)	4672	22592	44992	1664
Max time(ms)	464	1923	4220	238
Min time(ms)	342	1660	3439	162
Average time(ms)	380	1738	3556	188
gasUsed(wei)	1086576	5139464	10247921	404850

IX. CONCLUSION

In our work, we have proposed a blockchain-based EHR sharing scheme with conjunctive keyword searchable encryption and conditional proxy re-encryption to realize data security and privacy preservation of data sharing between different medical institutions. Firstly, we present a framework for EHR sharing among different entities based on cloud-assisted storage and blockchain. The cloud is in charge of storing EHR ciphertext while EHR indexes are kept on EHR consortium blockchain. Secondly, the network model, data structure and consensus mechanism for EHR consortium blockchain are designed to guarantee efficient operations of the system.

Moreover, we use keyword searchable encryption to ensure data security with searchability and employ conditional proxy re-encryption to realize data sharing with privacy preservation. Furthermore, we conduct security analysis and proof security of the proposed protocol, which demonstrates that our scheme can achieve the designed security goals. We also implement the scheme on Ethereum platform and evaluate the performance of computational overhead and communication overhead.

For future work, we will implement the scheme on Hyperledger Fabric and perfect smart contracts for running the algorithms of data sharing.

APPENDIX

A. SECURITY PROOF OF THEOREM 1

Suppose that $\mathcal{A}$ is an outsider adversary with advantage ε in attacking the proposed protocol against IND-CR-CKA and $H_1(\cdot), H_2(\cdot)$ the role of random oracles. We build a challenger $\mathcal{C}$ who can compute the solution of the DLDH problem by playing game with $\mathcal{A}$ as follows.

- ***Setup***: Given the DLDH parameters $(P_1, P_2, P_3, Q_1, Q_2, Q_3)$ where $Q_1 = a_1P_1$, $Q_2 = a_2P_2$ and $Q_3 = (a_1 + a_2)P_3$ or z. Challenger $\mathcal{C}$ randomly chooses $x \in Z_q^*$ as the DP's private key and computes $y = xP_1$. Additionally, it picks a number $\alpha \in Z_q^*$ randomly and keeps it secretly. Then, it sends $\mathcal{A}$ the system parameters $params = (G_1, G_2, \hat{e}, H_1(\cdot), H_2(\cdot), P)$ and the public key $pk = y$ while the x is unknown for $\mathcal{A}$.
  H_1 ***queries***: $\mathcal{C}$ maintains a list of tuples (w_i, c_i, h_i, u_i) called H_{1-list} for responding the queries of H_1. $\mathcal{A}$ queries the random oracle H_1 at most q_m keyword. When receiving the queries, challenger $\mathcal{C}$ responds as follows:
  1) If the query w_i already in H_{1-list}, $\mathcal{C}$ responds $h_i = H_1(w_i)$. Otherwise, it generates a random $c_i \in \{0, 1\}$.
  2) If $c_i = 0$, $\mathcal{C}$ selects a random number $u_i \in Z_q^*$ and computes $h_i \leftarrow u_iP_1$. Otherwise, it sets $h_i \leftarrow u_iP_3$.
  3) $\mathcal{C}$ adds the tuples (w_i, c_i, h_i, u_i) to H_{1-list} and returns h_i to $\mathcal{A}$.

  H_2 ***queries***: $\mathcal{C}$ maintains a list of tuples (w_i, c_i, f_i, v_i) called H_{2-list} for responding the queries of H_2. When $\mathcal{A}$ queries the random oracle H_2, challenger $\mathcal{C}$ responds as follows:
  1) If the query w_i is already in H_{2-list}, $\mathcal{C}$ responds $f_i = H_2(w_i)$. Otherwise, it generates a random $c_i \in \{0, 1\}$.
  2) If $c_i = 0$, $\mathcal{C}$ selects a random number $v_i \in Z_q^*$ and computes $f_i \leftarrow v_iP_2$. It computes $v_i = \frac{u_i}{\alpha}$ and sets $f_i \leftarrow v_iP_3$.
  3) $\mathcal{C}$ adds the tuples (w_i, c_i, f_i, v_i) to H_{2-list} and returns f_i to $\mathcal{A}$.
- ***Phase 1***: $\mathcal{A}$ queries some keyword set to trapdoor oracle.
  Trapdoor queries: $\mathcal{A}$ adaptively queries a keyword set $Q_i = (\Omega_{i,1}, \cdots, \Omega_{i,t})$ to get a trapdoor T_{Q_i}. $\mathcal{C}$ responds as follows:
  1) $\mathcal{C}$ executes the above algorithms for responding H_1, H_2 queries to get two lists $(w_{i,j}, c_{i,j}, h_{i,j}, u_{i,j})$ and $(w_{i,j}, c_{i,j}, f_{i,j}, v_{i,j})$.
  2) if there is any $c_{i,j} = 1$ for $1 \le j \le n$, then $\mathcal{C}$ aborts. Otherwise, it picks a random number $z_i \in Z_q^*$ and computes $T_{Q_i} = (T_{Q_{i,1}}, T_{Q_{i,2}}, T_{Q_{i,3}})$ where $T_{Q_{i,1}} = z_iP_1$, $T_{Q_{i,2}} = z_i\left(\sum_{j=1}^{t} u_{i,j}\right)P_1$ and $T_{Q_{i,3}} = z_i\left(\sum_{j=1}^{t} v_{i,j}\right)P_1$.
- ***Challenge***: $\mathcal{A}$ outputs a target keyword set W^* and sends it to $\mathcal{C}$. $\mathcal{C}$ performs as follows:
  1) It chooses a keyword set W' randomly and sets $W_0 = W^*$ and $W_1 = W'$ where $W_0 = \{\Omega_{0,1}, \cdots, \Omega_{0,n}\}$, $W_1 = \{\Omega_{1,1}, \cdots, \Omega_{1,n}\}$. The only restriction is that the previous trapdoors can't distinguish W_0 and W_1.
  2) It picks a random number $\beta \in \{0, 1\}$ and queries all keywords of W_β to H_1, H_2 oracles. Then it gets lists $(w_{\beta,i}, c_{\beta,i}, h_{\beta,i}, u_{\beta,i})$ and $(w_{\beta,i}, c_{\beta,i}, f_{\beta,i}, v_{\beta,i})$.
  3) If there is any $c_{\beta,i} = 1$ for all i, it computes a challenge ciphertext $C'_w = (A, B, C_{\beta,1}, \cdots C_{\beta,n})$ where $A = a_1P_1$, $B = \alpha a_2P_2$ and $C_{\beta,i} = u_{\beta,i}Q_1 + \alpha v_{\beta,i}Q_2$ (if $c_{\beta,i} = 0$) or $C_{\beta,i} = u_{\beta,i}Q_3$ (if $c_{\beta,i} = 1$). Otherwise, it aborts.
  4) Returns (W_0, W_1, C'_w) to $\mathcal{A}$.
- ***Phase 2***: $\mathcal{A}$ performs trapdoor queries as in phase 1. The restriction is that the generated trapdoor is indistinguishable for W_0 and W_1.
- ***Guess***: $\mathcal{A}$ outputs a guess $\beta' \in \{0, 1\}$. The challenger $\mathcal{C}$ will win the game if $\beta = \beta'$, which means $Q_3 = (a_1 + a_2)P_3$. Otherwise, it will lose, which means $Q_3 = z$.
  In the guess phase, if $Q_3 = (a_1+a_2)P_3$, which means the challenge ciphertext is a valid encryption of the keyword set W_β. In this case, the view of $\mathcal{A}$ is the same as the view in a real attack and it must have probability $1/2+\varepsilon$ at least. Thus, the challenger $\mathcal{C}$'s advantage to solve the DLDH problem is $\varepsilon_1 \ge \varepsilon$.

B. SECURITY PROOF OF THEOREM 2

Suppose that $\mathcal{A}$ is an adversary with advantage ε' in attacking the proposed protocol against IND-CCA. We build a challenger $\mathcal{C}$ who can compute the solution of the m-DBDH problem by playing game with $\mathcal{A}$ as follows.

- ***Setup***: Given a security parameter k, challenger $\mathcal{C}$ generates the system parameters $params = (q, P, G_1, G_2, \hat{e}, \hat{h}, t_1, t_2, t_3, H)$ and sends it to $\mathcal{A}$, where $\hat{h} = \hat{e}(P, P)$, $t_1 = r_1 P$, $t_2 = r_2 P$, $t_3 = r_3 P$ for randomly choosing three numbers $r_1, r_2, r_3 \in Z_q^*$, $H : \{0, 1\}^* \rightarrow Z_q^*$ is a one-way collision-resistant hash function.
- ***Phase 1***: $\mathcal{A}$ makes some queries.
  Public key queries: challenger $\mathcal{C}$ randomly picks $x_k \in Z_q^*$ and $d_k \in \{0, 1\}$. If $d_k = 1$, $\mathcal{C}$ chooses $sk_k = x_k$

as the DP's private key and computes $pk_k = x_k P$ as public key. Otherwise, it computes $pk_k = x_k r_1 P$ which means the private key is $r_1 x_k$ and unknown to $\mathcal{C}$. Then, $\mathcal{C}$ sends pk_k to $\mathcal{A}$ and stores (pk_k, x_k, d_k) in table T_k, where all public keys are recorded in T_k during the game.

Private key queries: $\mathcal{C}$ visits the table T_k and responds as follows: if $d_k = 1$, it returns $sk_k = x_k$ to $\mathcal{A}$. Otherwise, it outputs a random number in Z_q^* and aborts.

Re-encryption key queries: $\mathcal{C}$ visits the table T_k to find DP's public key (X_k, x_k, d_k), DO's public key (pk_i, x_i, d_i) and DR's public key (pk_j, x_j, d_j). Additionally, it receives the data packet $\vartheta_4 = (H_3(\Omega_i) \parallel F_i)$ and responds to $\mathcal{A}$ as follows:

1) If $d_k = 1$, it computes the re-encryption key $rk = (rk_1, rk_2) = (\frac{1}{x_i}(\sum_{i=1}^{t} H_3(\Omega_i))X_k, \frac{1}{x_i} pk_j H_3(F_i))$.
2) If $(d_k, d_i, d_j) = (0, 0, 0)$, it computes the re-encryption key $rk = (rk_1, rk_2) = (\frac{x_k}{x_i}P, \frac{x_j}{x_i}P)$.
3) If $d_k = 0$ and $(d_i, d_j) \neq (0, 0)$, it aborts.

Re-encryption queries: According to the result of re-encryption key queries, $\mathcal{C}$ generates the re-encryption ciphertext as in the section protocol description or aborts.

Decryption queries: $\mathcal{C}$ parses C'_m as $(c'_1, c'_2, c'_3, c'_4, c'_5, c'_6)$ at first. Then, it visits the table T_k to find (pk_j, x_j, d_j). If $d_j = 1$, it computes $m = \frac{c'_2}{(c'_3)^{1/sk_j H_3(F_i)}}$ and checks the equalities $\hat{e}(c'_4, P) = \hat{e}(c'_6, H_3(m)P_1 + P_2)$ and $\hat{e}(c'_1, P) = \hat{e}(c'_6, \sum_{i=1}^{t} H_3(w_i)P)$. If the equalities hold, $\mathcal{C}$ returns m to $\mathcal{A}$. Otherwise, it returns $\perp$.

- ***Challenge***: $\mathcal{A}$ presents two messages $m_0, m_1 \in G_2$, a target DP's public key pk'_k and a target DR's public key pk'_j. The restrictions are that $\mathcal{A}$ can't do any private key queries either on pk'_k or pk'_j in phase 1 and $\mathcal{A}$ can't do any re-encryption key queries.
  $\mathcal{C}$ picks a random number $\gamma \in \{0, 1\}$ and searches T_k to get (pk'_k, x'_k, d'_k) and (pk'_j, x'_j, d'_j). Then, it computes challenge ciphertext as follows:

$$\begin{aligned} C^*_m &= (c^*_1, c^*_2, c^*_3, c^*_4, c^*_5, c^*_6) \\ &= (b(\sum_{i=1}^{t} H_3(w_i))P, m_\gamma T^{\frac{1}{x'_k}}, \frac{x'_j}{x'_k} c^*_6, (r_1 H_3(m_\gamma) + r_2)c^*_6, \\ &(r_1 H_3(c^*_1) + r_2 H_3(c^*_1 \parallel c^*_2 \parallel c^*_4) + r_3)c^*_6, bP) \end{aligned}$$

  Finally, $\mathcal{C}$ returns challenge ciphertext C^*_m to $\mathcal{A}$.
- ***Phase 2***: $\mathcal{A}$ performs more queries as in phase 1. The restriction are as follows:
  1) $\mathcal{A}$ can't do any private key queries either on pk'_k or pk'_j as in phase 1.
  2) If $\mathcal{A}$ has made re-encryption key queries or re-encryption queries, it can't do private key queries anymore. Perhaps, if $\mathcal{A}$ has made a private key queries, then it can't make re-encryption key queries or re-encryption queries.
  3) $\mathcal{A}$ can't launch a decryption queries on target challenge ciphertext.
- ***Guess***: $\mathcal{A}$ outputs a guess $\gamma' \in \{0, 1\}$. The challenger $\mathcal{C}$ will win the game if $\gamma = \gamma'$, which means $T = \hat{e}(P, P)^{d/c}$. Otherwise, it will lose, which means $T \in G_2$ is a random number.
  Obviously, if $T = \hat{e}(P, P)^{d/c}$, which means the adversary can break the scheme with the advantage ε'. In this case, the view of $\mathcal{A}$ is the same as the view in a real attack and it must have probability $1/2+\varepsilon$ at least. Thus, the challenger $\mathcal{C}$'s advantage to solve the DLDH problem is $\varepsilon_2 \geq \varepsilon'$.

REFERENCES

[1] H. K. Patil and R. Seshadri, "Big data security and privacy issues in healthcare," in *Proc. IEEE Int. Congr. Big Data*, Anchorage, AK, USA, Jun./Jul. 2014, pp. 762–765.
[2] J. Li and X. Li, "Privacy preserving data analysis in mental health researc," in *Proc. IEEE Int. Congr. Big Data*, New York, NY, USA, Jun./Jul. 2015, pp. 95–101.
[3] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of Personal Health Records in cloud computing: Ciphertext-policy attribute-based signcryption," *Future Gener. Comput. Syst.*, vol. 52, pp. 67–76, Nov. 2015.
[4] X. Liu, Y. Xia, W. Yang, and F. L. Yang, "Secure and efficient querying over personal health records in cloud computing," *Neuro Comput.*, vol. 274, pp. 99–105, Jan. 2018.
[5] X. Liu, Q. Liu, T. Peng, and J. Wu, "Dynamic access policy in cloud-based personal health record (PHR) systems," *Inf. Sci.*, vol. 379, pp. 62–81, Feb. 2017.
[6] Z. Liu, J. Weng, J. Li, J. Yang, C. Fu, and C. F. Jia, "Cloud-based electronic health record system supporting fuzzy keyword search," *Soft Comput.*, vol. 20, pp. 3243–3255, Aug. 2016.
[7] X. Wang, A. Zhang, X. Ye, and X. Xie, "Secure-aware and privacy-preserving electronic health record searching in cloud environment," *Int. J. Commun. Syst.*, vol. 32, p. e3925, May 2019. doi: 10.1002/dac.3925.
[8] M. H. Au, T. H. Yuen, J. K. Liu, W. Susilo, X. Huang, Y. Xiang, and Z. L. Jiang, "A general framework for secure sharing of personal health records in cloud system," *J. Comput. Syst. Sci.*, vol. 90, pp. 46–62, Dec. 2017.
[9] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, vol. 90, May 2015, pp. 180–184.
[10] S. Amofa, E. B. Sifah, K. O.-B. Agyekum, S. Abla, Q. Xia, J. C. Gee, and J. B. Gao, "A blockchain-based architecture framework for secure sharing of personal health data," in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*. Ostrava, Czech Republic, 2018, pp. 1–6.
[11] X. Zheng, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Mere, "Blockchain-based personal health data sharing system using cloud storage," in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*. Ostrava, Czech Republic, Sep. 2018, pp. 1–6.
[12] T. Mikula and R. H. Jacobsen, "Identity and access management with blockchain in electronic healthcare records," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Prague, Czech Republic, 2018, pp. 699–706.
[13] S. Cao, G. Zhang, P. Liu, X. Zhang, and F. Neri, "Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain," *Inf. Sci.*, vol. 485, pp. 427–440, Jun. 2019.
[14] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.
[15] T. T. Thwin and S. Vasupongayya, "Blockchain based secret-data sharing model for personal health record system," in *Proc. 5th Int. Conf. Adv. Inform., Concept Theory Appl. (ICAICTA)*, Krabi, Thailand, 2018, pp. 196–201.
[16] A. Theodouli, S. Arakliotis, K. Moschou, K. Votis, and D. Tzovaras, "On the design of a blockchain-based system to facilitate healthcare data sharing," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun., 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, New York, NY, USA, Aug. 2018, pp. 1374–1379.
[17] L. X. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Gener. Comput. Syst.*, vol. 95, pp. 420–429, Jun. 2019.

[18] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.
[19] Y. Chen, S. Ding, Z. Xu, H. D. Zheng, and S. L. Yang, "Blockchain-based medical records secure storage and medical service framework," *J. Med. Syst.*, vol. 43, no. 5, Jan. 2019. doi: 10.1007/s10916-018-1121-4.
[20] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Pairing-Based Cryptography—Pairing* (Lecture Notes in Computer Science), vol. 4575. Berlin, Germany: Springer, 2007, pp. 2–22.
[21] P. Zeng and K.-K. R. Choo, "A new kind of conditional proxy re-encryption for secure cloud storage," *IEEE Access*, vol. 6, pp. 70017–70024, 2018.
[22] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of Cryptocurrency systems," in *Proc. 14th Annu. Conf. Privacy, Secur. Trust (PST)*, Auckland, New Zealand, 2016, pp. 745–752.
[23] D. Patel, J. Bothra, and V. Patel, "Blockchain exhumed," in *Proc. ISEA Asia Secur. Privacy (ISEASP)*, Surat, India, 2017, pp. 1–12.
[24] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Honolulu, HI, USA, Jun. 2017, pp. 557–564.
[25] W. B. Wang, D. T. Hoang, P. Z. Hu, Z. H. Xiong, D. Niyato, P. Wang, and Y. G. Wen, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2018. doi: 10.1109/ACCESS.2019.2896108.
[26] Y. Luo, Y. Chen, Q. Chen, and Q. Liang, "A new election algorithm for DPos consensus mechanism in blockchain," in *Proc. 7th Int. Conf. Digit. Home (ICDH)*, Guilin, China, 2018, pp. 116–120.
[27] T. Crain, V. Gramoli, M. Larrea, and M. Raynal, "DBFT: Efficient byzantine consensus with a weak coordinator and its application to consortium blockchains," 2017, *arXiv:1702.03068*. [Online]. Available: https://arxiv.org/abs/1702.03068
[28] K. Li, H. Li, H. Hou, K. Li, and Y. Chen, "Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain," in *Proc. IEEE 19th Int. Conf. High Perform. Comput. Commun., IEEE 15th Int. Conf. Smart City, IEEE 3rd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Bangkok, Thailand, Dec. 2017, pp. 466–473.
[29] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *J. Med. Syst.*, vol. 42, p. 140, Aug. 2018. doi: 10.1007/s10916-018-0995-5.

YONG WANG received the B.E. degree in electronic and information engineering from Anhui Normal University, Wuhu, China, in 2013, where he is currently pursuing the M.S. degree in Internet of Things technology with the School of Physical and Electronic Information Engineering. His research interests include applied cryptography and healthcare blockchain.

AIQING ZHANG received the M.S. degree in circuits and systems from Xiamen University, China, in 2006, and the Ph.D. degree in signal and information processing from the Nanjing University of Posts and Telecommunications, China, in 2016. She is currently a Professor with Anhui Normal University, China. She has authored more than 30 articles, and holds more than ten inventions. Her research interests include blockchain, applied cryptography, and wireless network security.

PEIYUN ZHANG received the B.S. degree in applied electronics from Anhui Normal University, Wuhu, China, in 1998, the M.S. degree in computer science and technology from Northwest University, Xi'an, China, in 2005, and the Ph.D. degree in computer science and technology from the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China, in 2008. She was a Postdoctoral Researcher with the University of Science and Technology China, Hefei, China, from 2010 to 2013, and a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, in 2016. She is currently a Professor with the School of Computer and Information, Anhui Normal University. Her current research interests include blockchain, cloud computing, big data, trust computing, Petri nets, web service, and intelligent information processing. She has published over 50 articles in the above areas.

HUAQUN WANG received the B.S. degree in mathematics education from the Shandong Normal University, China, the M.S. degree in applied mathematics from the East China Normal University, China, in 1997 and 2000, respectively, and the Ph.D. degree in cryptography from the Nanjing University of Posts and Telecommunications, in 2006, where he is currently a Professor. His research interests include applied cryptography, network security, and cloud computing security.