IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 5, 2020, accepted March 17, 2020, date of publication March 24, 2020, date of current version April 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982964

Blockchain-Based Secure Storage and Access Scheme For Electronic Medical Records in IPFS

JIN SUN, XIAOMIN YAO, SHANGPING WANG, AND YING WU
School of Science, Xi'an University of Technology, Xi' an 710054, China

Corresponding author: Xiaomin Yao (1556259441@qq.com)

This work was supported in part by the National Natural Science Youth Foundation of China under Grant 61303223, and in part by the National Natural Science Foundation of China under Grant 61572019.

ABSTRACT Electronic medical records can help people prevent diseases, improve cure rates, provide a significant basis for medical institutions and pharmaceutical companies, and provide legal evidence for medical negligence and medical disputes. However, the integrity and security problems of electronic medical data still intractable. In this paper, based on the ciphertext policy attribute-based encryption system and IPFS storage environment, combined with blockchain technology, we constructed an attribute-based encryption scheme for secure storage and efficient sharing of electronic medical records in IPFS storage environment. Our scheme is based on ciphertext policy attribute encryption, which effectively controls the access of electronic medical data without affecting efficient retrieval. Meanwhile, we store the encrypted electronic medical data in the decentralized InterPlanetary File System (IPFS), which not only ensures the security of the storage platform but also solves the problem of the single point of failure. Besides, we leverage the non-tamperable and traceable nature of blockchain technology to achieve secure storage and search for medical data. The security proof shows that our scheme achieves selective security for the choose keyword attacks. Performance analysis and real data set simulation experiments shows that our scheme is efficient and feasible.

INDEX TERMS Access control, attribute-based encryption, blockchain, electronic medical records, InterPlanetary File System (IPFS).

I. INTRODUCTION

Currently, the rapid development of information technology has made electronic information systems more widely used in medical treatment, and a large amount of medical data is generated every day, such as electronic medical records, medical images, diagnostic reports, infectious diseases, etc., and proper leverage of these medical data not only can infectious diseases be predicted in advance, and prepare for protection, but it can also be used as a legal evidence for medical disputes. Therefore, how to efficiently leverage medical data is worthy to study.

Electronic medical data can reflect the treatment situation of patients promptly on time, and share treatment experience with other medical institutions. However, once the shared medical data are abused illegally, the patient's privacy will be leaked. Therefore, controlling the access right of medical data is an urgent issue. Currently, attribute-based encryption (ABE)is the best way to implement access control [1]–[3]. Medical institutions outsource encrypted medical data to third-party (i.e. Cloud server) for management, which not only helps reduce the computational cost, saves local storage space, but also provides efficient retrieval of data. In addition, applying blockchain technology to modern medical scenarios has become a new trend [4], [5], however, the medical data cannot be efficiently retrieved which stored on the blockchain. Literature [6] and [7] combines the blockchain and the cloud server to deal with how to retrieve the data while how to provide a safeguard for the data security, but since the cloud server is centralized, once the single cloud model fails, it will lead to the whole cloud server is not available.

To solve this problem, we consider InterPlanetary File System (IPFS) as our storage platform. IPFS is a decentralized storage protocol designed to address excessive file redundancy, and it allocates a unique hash for each stored file, the user can find the corresponding file according to the hash address. Since IPFS is decentralized, there is no single

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

point of failure. Before storing data, we encrypt the medical data based on attributes and determine the attributes of users (including doctors, nurses, patients, researchers, etc.) [8]. The user's private key is related to their attributes, while the ciphertext is related to the policy. The user can decrypt the ciphertext if and only if the user's private key meets the access policy in the ciphertext. Also, we can use the blockchain to record the storage and search process of medical data, which can not only track the source of the data but also record the data retrieval process. Furthermore, we store the hash value of medical data in the blockchain, which provides strong evidence for the originality of the user's verification data.

Our Contribution: In order to address the effective access control of electronic medical records and the semi-honest and curious question of cloud servers, we constructed an attribute-based encryption scheme for secure storage and access of electronic medical records in the IPFS storage environment. Our scheme leverage IPFS as a storage platform and the blockchain records the entire process of data storage and search, which provides strong evidence for medical disputes and medical negligence. Our scheme has the following advantages:

A. SECURE CONTENT STORAGE

Our scheme stores medical data on a decentralized Inter-Planetary file system (IPFS) rather than a semi-honest and curious cloud server, which further protects the privacy of medical data; Besides, IPFS allocates a unique hash for each file, therefore, files can not be stored repeatedly while saving storage space to a certain extent.

B. VERIFIABLE KEYWORD SEARCH

In our scheme, on the one hand, users can quickly find the corresponding medical records according to the specified keywords; on the other hand, users can also verify the originality of the medical records and track the source of the medical records.

C. ACCESS CONTROL

In order to prevent the unrelated person from viewing the patient's medical history, we assign different access rights based on user's attribute. The user can decrypt the ciphertext if and only if the user's private key satisfies the access policy in the ciphertext.

The rest of this paper is arranged as follows: In section II, we introduce the related work of our scheme. In section III, we present the background knowledge about our scheme. Section IV gives the system model and security model of our scheme, Section V is the specific details of our scheme. In Section VI, we show the security of the scheme and section VII analyzed the effectiveness and feasibility of our scheme. At last, we draw the conclusion of this whole paper.

II. RELATED WORKS

Attribute-based encryption (ABE) can be separated into the key-policy attribute-based encryption(KP-ABE) [9] and the ciphertext-policy attribute-based encryption (CP-ABE) [10]. KP-ABE is mostly used in biometric identification systems [8]. CP-ABE is mostly used for encrypted storage systems. Due to attribute-based encryption provides secure fine-grained access control for data, thus, it is being applied more and more to various scenes. The most common is to encrypt medical data based on attribute and outsource it to cloud servers for management, which is not only saving local storage costs but also ensuring data security. Sun *et al.* [11] proposed an attribute-based keyword search scheme with effective user revocation (ABKS-UR), enabling scalable fine-grained search authorization, allowing multiple owners to separately encrypt and outsourcing their data to the cloud server. And by combining proxy re-encryption and lazy re-encryption technology, the heavy system update workload is delegated to the resource-rich semi-trust cloud servers during user revocation. However, in this scheme, it is impossible to verify whether the cloud server has returned all relevant results honestly, and also cannot verify their correctness. Guo *et al.* [12] proposed an access control architecture named EHR for controlling access of electronic health records which are stored in the semi-honest and curious cloud server, they leverage CP-ABE technique to encrypt medical data and assign different users with distinct right to search. However, the correctness of lots of search results cannot be verified, while a great deal of not corresponding results returned by the server will waste resources. Su *et al.* [13] proposed an attribute-based encryption scheme in the cloud environment, which not only enables the server to decrease the number of encryption and decryption of the attribute, but also effectively protects the security of the data, and achieves effective to control data access; the VKSE scheme is proposed by Miao *et al*. [14] uses the access tree to implement the access policy, which achieved fine-grained access control of data meanwhile supporting data search and verification. However, all the above schemes store data on a semi-honest and curious cloud server. Once a single cloud model failure, the entire cloud servers will be unavailable. Li *et al.* [15] proposed an extended encryption scheme based on hierarchy attributes of files, which can encrypt multiple files at the same access level, enabling users in cloud storage to realize secure and flexible access control, however, they solve the security issues for central authority by using attribute encryption, which makes their programs inefficient and unable to adapt to large multi-sector organizations and companies.

The medical record is important data in the medical system, and a perfect medical record is critical for the patient's return to visit. Currently, some medical institutions use the attribute to encrypt electronic medical records and outsource them to the cloud servers for management. Wang and Lin [16] solve the problem of efficiency and security existing in accessing control for personal health records (PHRs) through the mobile client in the environment of the Cloud Storage, they put forward a scheme for mobile applications' access policy of PHRs under a semi-trusted server framework, and handle the efficient and on-demand user revocation by

optimizing the existing MA-ABE program and introducing lazy re-encryption and proxy re-encryption technology, but there is no guarantee whether lazy re-encryption is secure. Alshehri *et al.* [17] proposed a scheme for secure access to medical data in a cloud environment by using ciphertext-policy attribute-based encryption, which provides effective control of access to medical data, but in this scheme, the medical data stored in the cloud cannot be retrieved. Subsequently, Xu *et al.* [18] proposed a dynamic medical data storage scheme that supports the insertion and deletion of medical data, but in this system, the originality and accuracy of the searched data cannot be verified.

These schemes store all of the encrypted data on a cloud server. However, the cloud server is centralized, once a single cloud model is a failure, the entire cloud servers will be paralyzed and all data will be inaccessible. The InterPlanetary File System (IPFS) is a decentralized storage protocol designed to solve the problem of excessive file redundancy. It has the following advantages: (a)IPFS allocates a unique hash value according to the file content instead of the location stored the file as its acquisition path, which prevents IPFS from repeatedly storing the same file and saving storage space. (b)IPFS is a decentralized storage protocol, which can permanently save and share various types of files. (c)For high-frequency request data, IPFS will create duplicate data according to the previous request path. In the next request, these data can be read directly locally [19]. Therefore, IPFS is more suitable as a storage platform for medical data than cloud servers. Chen *et al.* [20] proposed an improved P2P file system scheme based on IPFS and blockchain, which present the role of content service providers to solve the high-throughput issue of single users in IPFS. Zheng *et al.* [19] proposed an innovative IPFS-based blockchain storage model, in which miners store a large number of files in IPFS and store the returned IPFS hash in blocks, Thereby reducing the load on the blockchain. Inspired by the above literature, we put forward a scheme for encrypting electronic medical records based on attributes, in which data is stored in IPFS and the entire process of storage and search is recorded by the blockchain.

III. PRELIMINARIES

In table 1, we give an explanation of the symbols which will be used in our system.

A. BILINEAR MAP

Definition 1 (Bilinear Map [13]): Let G_1, G_2 and G_T be three multiplicative cyclic groups of the prime order p, g_1, g_2 be the generators of G_1 and G_2, $e : G_1 \times G_2 \longrightarrow G_T$ be the bilinear map which has several properties:

- Bilinearity: $\forall g_1 \in G_1, g_2 \in G_2, \forall a, b \in Z_p^*, e\left(g_1^a, g_2^b\right) = e\left(g_1, g_2\right)^{ab}$.
- Non-degeneracy: $\exists g_1 \in G_1, g_2 \in G_2, e\left(g_1, g_2\right) \neq 1$.
- Computability: $\forall g_1 \in G_1, g_2 \in G_2, e\left(g_1, g_2\right)$ can be effectively calculated.

TABLE 1. Symbols table.

Symbol	Description
R	electronic medical record
Sig_R	signature of electronic medical record
I_{wj}	index of electronic medical record
H	hash address returned by IPFS
h_R	hash of electronic medical record
h_I	hash of index for electronic medical record
(T_1, T_2)	access request
W	access policy
S	attribute set of data user
ID	block number on the blockchain
ST_w	search token

B. HARDNESS ASSUMPTIONS

Definition 2 (Asymmetric Decision Bilinear Diffie-Hellman (DBDH) Assumption [16]): Given the bilinear map parameters $(G_1, G_2, p, g_1, g_2, e)$ and elements $z_1, z_2, z_3, Z \in Z_p^*$ and $g_1 \in G_1, g_1 \in G_2$ be generators, the DBDH assumption is that the advantage of adversary in distinguishing the tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_2}, g_2^{z_3}, e\left(g_1, g_2\right)^{z_1 z_2 z_3}\right)$ from the tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_2}, g_2^{z_3}, e\left(g_1, g_2\right)^{Z}\right)$ is negligible, where the adversary $'$s advantage is defined as

$$
\begin{aligned}
&Adv_A^{DBDH}\left(1^k\right) \\
&= |Pr\left[A\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_2}, g_2^{z_3}, e\left(g_1, g_2\right)^{z_1 z_2 z_3}\right) = 1\right] \\
&\quad - Pr\left[A\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_2}, g_2^{z_3}, e\left(g_1, g_2\right)^{Z}\right) = 1\right]|
\end{aligned}
$$

Definition 3 (Decision Linear Assumption) [21]: Given the parameters $(G_1, G_2, p, g_1, g_2, e)$ and elements z_1, z_2, z_3, $z_4, Z \in Z_p^*$ be chosen at random and $g_1 \in G_1$, $g_2 \in G_2$ be generators. The D-Linear assumption named that the advantage of adversary in distinguishing the tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1}, g_1^{z_1 z_3}, g_1^{z_2 z_4}, g_2^{z_1 z_3}, g_1^{z_3+z_4}\right)$ from the tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1}, g_1^{z_1 z_3}, Z, g_2^{z_1 z_3}, g_1^{z_3+z_4}\right)$ is negligible, where the adversary $'$s advantage is defined as

$$
\begin{aligned}
&Adv_A^{D-L}\left(1^k\right) \\
&= |Pr\left[A\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1}, g_1^{z_1 z_3}, g_1^{z_2 z_4}, g_2^{z_1 z_3}, g_1^{z_3+z_4}\right) = 1\right] \\
&\quad - Pr\left[A\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1}, g_1^{z_1 z_3}, Z, g_2^{z_1 z_3}, g_1^{z_3+z_4}\right) = 1\right]|
\end{aligned}
$$

Definition 4 (Asymmetric Decision Diffie-Hellman (DDH 1v1) Assumption [13]): Given the bilinear map parameters $(G_1, G_2, p, g_1, g_2, e)$ and elements $z_1, z_2, Z \in Z_p^*$, the DDH 1v1 assumption states that the advantage of adversary in distinguishing the tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1 z_2}\right)$ from the tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{Z}\right)$ is negligible, where the adversary $'$s advantage is defined as

$$
\begin{aligned}
Adv_A^{DDH1v1}\left(1^k\right) &= |Pr\left[A\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1 z_2}\right) = 1\right] \\
&\quad - Pr\left[A\left(g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{Z}\right) = 1\right]|
\end{aligned}
$$

C. ACCESS STRUCTURE

Threshold structure [8], tree-based access structure [10], [14], AND-gates [22] and linear secret sharing structure [23], [24]

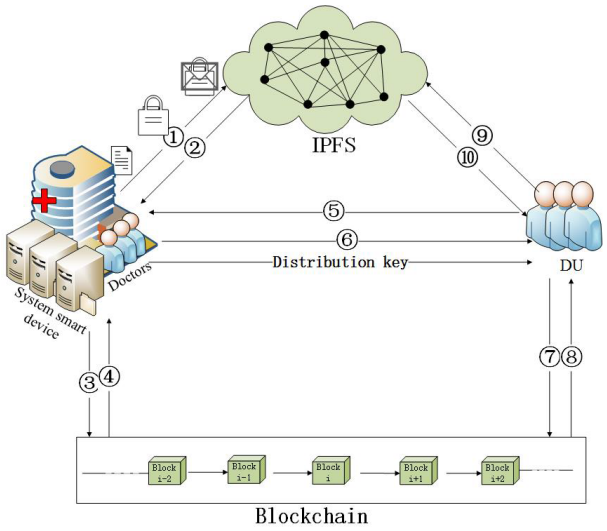

FIGURE 1. System model.

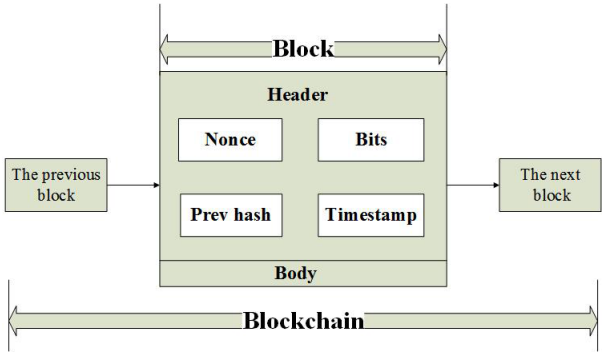

FIGURE 2. Blockchain.

is often used to achieve access control in attribute-based encryption scheme. Here, we exploit a series of AND-gates with multi-valued attributes like [25] as our access structure.

Definition 5 (Access Structure [21]):

Let $U = \{att_1, att_2, \cdots, att_m\}$ be a set of attributes. For each $att_i \in U$, $L_i = \{v_{i,1}, v_{i,2}, \cdots v_{i,m_i}\}$ is a set of possible values, where m_i is the total of possible values for att_i. Then let $S = \{S_1, S_2, \cdots, S_m\}$ be an attribute set of a user, where $S_i \in L_i$ and $W = (W_1, W_2, \cdots, W_m)$ be an access structure where $W_i \in L_i$. Note that $S \vDash W$, if the attribute set S meets the access structure W, namely $S_i = W_i, i = (1, 2, \cdots, m)$.

IV. PROBLEM FORMULATIONS

A. SYSTEM MODEL

There exist four main entities in our scheme, namely Blockchain, InterPlanetary File System (IPFS), medical system (including Doctors and system smart device), and data user (DU). The data generated by the medical system will be stored in the IPFS while ensuring its privacy, searchability, and verifiability. The main processes are shown in Figure 1.

1) BLOCKCHAIN

Blockchain is a novel application model that integrates decentralized data storage, peer-to-peer transmission, consensus mechanism, encryption algorithm, and other technologies. Here, we leverage blockchain to record the storage and search process of medical data. Due to the immutability of the blockchain, the data stored on the blockchain cannot be arbitrarily modified. So it can be used as evidence for verifying the originality and fluidity of the data.

2) IPFS

The InterPlanetary File System (IPFS) is a decentralized storage protocol. It can permanently store and share various types of files, and it will allocate a unique hash value based on the file content, so we can easily find the files according to the hash value. Besides, IPFS has a deduplication mechanism, which can effectively avoid repeating storage of data and saves the storage space. In this paper, we leverage IPFS to store our medical records.

3) MEDICAL SYSTEM

The medical system includes doctors and system smart device, doctors in our system is responsible for encrypting and saving the medical records. He first runs the Encrypt algorithms and Index algorithms to encrypt the medical records and generate indexes for the keywords. Then the ciphertext is stored in the IPFS, and finally, the hash of the medical records and the hash address returned by the IPFS are recorded on the blockchain. The system smart device is primarily responsible for the registration of all personnel in the hospital, and their attributes (i.e. physicians, nurses, patients, researchers, etc.) identificaon, besides, it also assigns secret private key associated with their attribute for registered personnel.

4) DATA USER(DU)

The data user (DU) can be a doctor, a nurse, a researcher, a patient, etc. Each DU can get a key pair assigned by the system, which is associated with their attributes. In order to search for a medical record, DU first needs to make a request to the hospital. If the system smart device affirms that their attributes meet the access policies, the system smart device will return a token for them to search for the medical records he needed.

5) WORK FLOW

For easy to understand, in our scheme, we instantiate a provably secure storage and access scheme for electronic medical records. Specifically, after a patient goes to the hospital system to register, the doctor diagnoses the condition and generates a medical record. The doctor needs to save the records so that the patient can return to visit his data and the researcher can learn some details at any moment. Since the medical records are related to the patient's personal privacy, the medical records must be encrypted before saving. The doctor encrypts and signs the medical records,

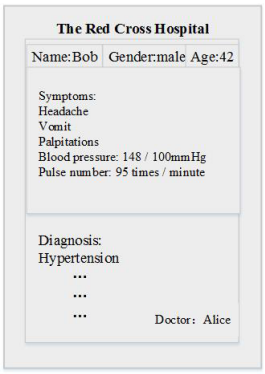

FIGURE 3. The medical record of Bob.

then uploads it to the IPFS for storage, and generates indexes for keywords, as shown in step 1. IPFS returns a hash address of the stored file to the doctor, as shown in step 2. After receiving the hash address, doctor encrypts the hash address with a random number, and hashes the medical records and its index with SHA256 hash function, then stores the hash value and the encrypted hash address on the blockchain by broadcasting a transaction, as shown in step 3. The blockchain will return a transaction ID, as shown in step 4. Upon receiving the transaction ID from blockchain, the data storage phase is completed. When the patient returns next time to visit his record, he first needs to call up the previous medical records. DU sends an access request containing keywords to the hospital, the system smart device determines whether the DU has access rights and returns a search token to the DU after he confirms, as shown in steps 5 and 6. The DU can read the corresponding hash value on the blockchain according to the block ID contained in the search token, and verify whether the hash address contained in the token was tramped by comparing with the hash address on the blockchain, as shown in steps 7 and 8. In steps 9 and10, DU downloads the ciphertext in the IPFS using the verified hash address. Besides, DU decrypts the ciphertext with its own secure private key and obtains the original medical records. Finally, DU verifies whether it is consistent with the hash value on the blockchain.

For example, Alice is a doctor in the Red Cross Hospital, Bob is a patient, and he wants to go to the hospital. He first registered at the hospital, then the system smart device identifies its attributes as $S = \{name : Bob, age : 42, gender : male, identify : patient\}$, and generates a public-private key pair for Bob based on its attributes. Dr. Alice generates a medical record R for Bob after Bob's diagnosis, the medical record R is shown in figure3:

Alice first encrypts the medical record R and gets ciphertext CT. Secondly, Alice signs the ciphertext CT and gets sig_R. And thirdly, Alice stores $CT^* = (CT, sig_R)$ in the IPFS while IPFS returns a hash address h about the CT^*. Finally, Alice stores the tuple (h_R, h') on the blockchain by broadcasting a transaction and obtains the block ID, where h_R is the hash value of the medical record, and h' is the encrypted hash address. For convenience of search, Alice extracts the keyword $w = \{Bob, male, 42, hypertension\}$ from the medical record R, and then generates an index $\{I_{w_j}\}_{j\in[1,m]}$ for keyword w. When Bob returns to visit, he first needs to call up the previous medical record R. He fist sends a request to hospital containing the keyword $w' = \{Bob, male\}$. The system smart device returns a search token $ST_w = (ID, h, \gamma)$ after it verifies the identity of the Bob. Bob first verifies the hash address h which contains in the search token ST_w, and then he downloads the encrypted medical record CT^* from the IPFS according to the hash address h. Finally he can get the medical record after decrypting with his private key.

B. OUR SCHEME OVERVIEW

Our scheme contains eight algorithms as follows:

•**Init**$(1^k) \longrightarrow (mpk, msk)$: System initializes and gets the system public key mpk and the master secret key msk.

•**KeyGen**$(msk, S) \longrightarrow (pk_o, sk_o), (sk_u, sk_S)$: The system runs KeyGen algorithm by inputting a master key msk and an attribute set S of the DU, then it returns the public-private key pair (pk_o, sk_o) of Doctor and the private key pair (sk_u, sk_S) of DU, where sk_S is the private key associated with set of their attribute, sk_u is the search private key.

• **Encrypt**$(pk_o, R) \longrightarrow (CT)$: Doctor runs Encrypt algorithm, it inputs the public key pk_o and medical record R, then returns a ciphertext set CT. Doctor stores the ciphertext set CT in IPFS, and generates hash for the file and stores it on the blockchain by broadcasting a transaction.

• **Index**$(pk_o, w) \longrightarrow I$: Doctor first extracts the keyword set w from the medical records R, and then runs the Index algorithm, he inputs the public key pk_o and keyword set w, and returns index $\{I_{w_j}\}_{j\in[1,m]}$ finally.

• **Search**$(T_1, T_2) \longrightarrow ST_w$: The DU generates a search request (T_1, T_2) for target keywords and sends the request (T_1, T_2) to hospital. If the request (T_1, T_2) matches the index I, then system smart device returns a search token ST_w to DU.

• **Verify 1**: Upon the DU receives the search token ST_w, he first gets the file hash and encrypted hash address from the blockchain based on the ID in the token ST_w, then encrypts the hash address with the random number which contains in the search token, to verify whether it is consistent with the encrypted hash address on the blockchain.

• **Decrypt**$(CT, sk_u) \longrightarrow R$: The DU gets ciphertext according to the hash address, and then runs Decrypt algorithm, she inputs the private key sk_S and ciphertext CT, then he obtains the original medical records.

• **Verify 2** : The DU hashes the decrypted medical record R' with SHA256 hash function, and then he compares the hash value h'_R with the hash value h_R obtained from the blockchain. If $h'_R = h_R$, it shows that the data is original and has not been tampered with.

C. SECURITY MODEL

We use the following games to describe the security model of our scheme. Our scheme is selectively secure if no polynomial − time adversary can win the following games with a non-negligible advantage:

Selective Game for CP-ABE:

Init: Challenger gets two ciphertext policies W_0, W_1 promised by the adversary.

Setup: The challenger first inputs a security parameter λ and gets the system public parameter, then the adversary gets the public key pk_0 from challenger by running the KeyGen algorithm.

Phase 1: In the KeyGen query stage,if $S \vDash W_0 \wedge S \vDash W_1$ or $S \nvDash W_0 \wedge S \nvDash W_1$, where S is the attribute of adversary, then the adversary will gets the private key sk_S generated from their attribute. This phase can be performed multiple times.

challenge: The adversary begins the challenge stage by promising two records R_0 and R_1. If the adversary got the sk_S generated from their attribute S, meanwhile the attribute set S satisfies both policies W_0 and W_1 in phase 1,then it is means that $R_0 = R_1$. The challenger encrypts (sk_o, R_b, W_b) by flipping a random coin b. If $b = 0$, he encrypts (sk_o, R_0, W_0) and sends it to the adversary A, else, he encrypts (sk_o, R_1, W_1) and sends it to the adversary A.

Phase 2: Similar to Phase 1. If $R_0 \neq R_1$, the adversary cannot submit S such that $S \vDash W_0 \wedge S \vDash W_1$.

Guess: The adversary A makes a guess b', and the adversary's advantage in this game is $Adv = |Pr\left(b' = b\right) - \frac{1}{2}|$.

The selective keywords attack (CKA) **security games:**

Init: Challenger gets a ciphertext policy W^* promised by the adversary.

Setup: The challenger first inputs a security parameter λ and gets the system public parameter, and the adversary gets the public key pk_0 from challenger by running the KeyGen algorithm.

Phase 1: In the KeyGen query stage,If $S \vDash W_0 \wedge S \vDash W_1$ or $S \nvDash W_0 \wedge S \nvDash W_1$, where S is the attribute of adversary. Then the adversary will get the private key sk_u, This phase can be performed multiple times

challenge: The adversary begins the challenge algorithm by promising two keywords w_1 and w_2. The challenger encrypts w_b by flipping a coin. If $b = 0$, he encrypts w_1, otherwise he encrypts w_2. Then he sends the index to the adversary A.

Phase 2: Similar to Phase 1.

Guess: The adversary A makes a guess b', and the adversary's advantage in this game is $Adv = |Pr\left(b' = b\right) - \frac{1}{2}|$.

V. OUR CONCRETE CONSTRUCTION

In the medical system, registered users (i.e. doctors, nurses, patients, researchers, etc.) can upload and search medical records. In our scheme, we instantiate a process of a patient goes to hospital and then returns to visit.

Init: Given a security parameter λ, the system first outputs a system public parameter $(G_1, G_2, G_T, \lambda, e, p, g_1, g_2,)$, where G_1, G_2 are two multiplicative cyclic groups of the prime order p, g_1, g_2 are the generators of G_1, G_2 and $g_1 \in G_1, g_2 \in G_2$, $e : G_1 \times G_2 \rightarrow G_T$ are the bilinear maps. A collision resistant hash function is named as $H : \{0, 1\}^{\lambda} \rightarrow Z_p^*$. Then system randomly select $\alpha, \beta, \sigma \in Z_p^*$, for each attribute number $\{i|1 \leqslant i \leqslant m\}$, the system generated $\left\{a_{i,t} \mid 1 \leqslant t \leqslant m_i\right\} \in Z_p^*$ randomly, where m_i is the total of all possible values of i, (For example, when $i = 1$ indicates a doctor, $i = 2$ indicates a nurse, and $i = 3$ indicates an expert, then m_1 represents the total number of doctors, and m_3 represents the total number of experts); Finally it gets the public key $mpk = \{\mathcal{GP}, e(g_1, g_2), H\}$ and the master key $msk = \left\{g_1, g_2, \alpha, \beta, \sigma, \left\{a_{i,t}|1 \leqslant t \leqslant m_i, 1 \leqslant i \leqslant m\right\}\right\}$.

KeyGen: In this phase, The system generates the public-private key pair for the Doctor and the private key pair for the DU; Doctor encrypts the medical records with his public key and stores it in the IPFS; DU generates a search request with its own search private key, then decrypts the medical records with his secret private key which associate with the set of his attribute.

Setp1: Generating a public-private key pair for Doctor. The system first computes $Y = e(g_1, g_2)^{\alpha}$, $B = g_1^{\beta}$, $u_1 = g_1^{\sigma}$ and $A_{i,t} = g_1^{a_{i,t}}$, where $1 \leqslant t \leqslant m_i$, $1 \leqslant i \leqslant m$. Then he gets the public-private key pair (pk_o, sk_o) as following:

$$pk_o = \left\{Y, B, u_1, \left\{A_{i,t}|1 \leqslant t \leqslant m_i, 1 \leqslant i \leqslant m\right\}\right\}$$
$$sk_o = \left\{\alpha, \beta, \sigma, \left\{a_{i,t}|1 \leqslant t \leqslant m_i, 1 \leqslant i \leqslant m\right\}\right\}$$

Setp2: Generating a private key for DU with attribute $S = [S_1, S_2, \cdots, S_m] = [v_{1,t_1}, v_{2,t_2}, \cdots, v_{m,t_m}]$. The system randomly selects $s \in Z_p^*$, computing $D_1 = g_2^s$, $u_2 = g_2^{\sigma}$, and $D_2 = \left(u_2\, g_2^{\sum\limits_{\forall i,t \in s_i} a_{i,t}}\right)^s$; Besides, the system selects two random numbers $x_i, \lambda_i \in Z_p^*$, where $1 \leqslant i \leqslant m$ lets $x = \sum\limits_{i=1}^{m} x_i$, and computes $d_0 = g_2^{\frac{\alpha+x}{\beta}}$. For each attribute number $\{i|1 \leqslant i \leqslant m\}$, when $S_i = v_{i,t_i}$, he computes $d_{i,1} = g_2^{x_i + a_{i,t_i}\lambda_i}$, $d_{i,2} = g_2^{\lambda_i}$, then it gets the private key pair (sk_u, sk_S) as following:

$$sk_u = \{D_1, D_2\}$$
$$sk_S = \left\{d_0, \left\{d_{i,1}, d_{i,2}|1 \leqslant i \leqslant m\right\}\right\}$$

Encrypt: Doctor generates a medical record R for DU after he diagnoses the condition of DU, and then stores the medical record R on IPFS. Firstly, in order to encrypt the medical record R, doctor designations the ciphertext policy as $W = [W_1, W_2, \cdots, W_m]$, for each attribute number $\{i|1 \leqslant i \leqslant m\}$, he randomly selects $r_i \in Z_p^*$, and lets $r = \sum\limits_{i=1}^{m} r_i$. He computes $C' = R \cdot Y^r$ and $C_0 = B^r$ at first, then he computes $C_{i,1} = g_1^{r_i}$; If $v_{i,t} \in W_i$, the doctor lets $C_{i,t,2} = A_{i,t}^{r_i}$, otherwise, let $C_{i,t,2}$ be a random element; Then the ciphertext is

$$CT = \left\{C', C_0, \left\{C_{i,1}, C_{i,t,2}|1 \leqslant t \leqslant m_i, 1 \leqslant i \leqslant m\right\}\right\}$$

Secondly, the Doctor signs the ciphertext CT:

$$sig_R = \left(H\left(id_R\right) \cdot g_1^{H(CT)}\right)^{\sigma}$$

where id_R is the identifier of R.

Thirdly,the doctor stores $CT^* = (CT, sig_R)$ in the IPFS, and the IPFS returns a hash address h about CT^*;

Finally, doctor records the process of storing medical records on the blockchain: He first hashes medical record R with SHA256 hash function and gets $h_R = SHA_{256}\langle R\rangle$, for hash address h, he randomly selects $\gamma \in Z_p^*$, and computes $h' = h^\gamma$. Next he stores the tuple (h_R, h') on the blockchain by broadcasting a transaction and obtain the block ID.

Index: For the convenience of search, the Doctor needs to generate indexes for stored medical records. He first extracts the keyword $w = \{w_1, w_2, \cdots, w_m\}$ from the medical record R, then selects $y \in Z_p^*$ and computes $I_1 = g_1^{\overline{H(w_1\|w_2\|\cdots\|w_m)}^{y}}$, $I_2 = \left(u_1 \cdot \prod_{\forall i,t\in W_i} A_{i,t}\right)^y$. Thus he can get the index as $\{I_{w_j}\}_{j\in[1,m]} = (I_1, I_2)$.

Search: In this stage, DU wishes to search the keyword $w' = \{w'_1, w'_2, \cdots, w'_m\}$, the DU first sends a search request to the hospital. The system smart device returns a search token after it verifies the identity of the DU.

Request: DU sends a request to the hospital containing the keyword $w' = \{w'_1, w'_2, \cdots, w'_m\}$. He selects $z \in Z_p^*$ randomly, and computes $T_1 = D_1^{\overline{H(w'_1\|w'_2\|\cdots\|w'_m)}^{z}}$, $T_2 = D_2^z$. Then he sends the request (T_1, T_2) to the hospital.

Test: After receiving the request from the DU, the system smart device tests whether the DU has the right to view the medical records containing the keywords $w' = \{w'_1, w'_2, \cdots, w'_m\}$. The system smart device first judges whether $e(I_1, T_2)$ is equal to $e(I_2, T_1)$, if $(I_1, T_2) = e(I_2, T_1)$, it means that the DU has the right to access the medical records which containing the keywords $w' = \{w'_1, w'_2, \cdots, w'_m\}$. Then the system smart device returns a search token $ST_w = (ID, h, \gamma)$ to DU, in which contains a block ID, a hash address h and a random number γ.

In terms of the equation $e(I_1, T_2) = e(I_2, T_1)$, it is a matching of the request with the index. In the keyword index generation process, the doctor encrypts m keywords to get $\{I_{w_j}\}_{j\in[1,m]}$. In the request generation process, the DU sends a search request (T_1, T_2) containing m' keywords to the system smart device of the hospital, particularly $m' \leqslant m$. When the system smart device obtained the request (T_1, T_2), in order to hold the above equation, the system smart device randomly choose m' index from $\{I_{w_j}\}_{j\in[1,m]}$ and perform multiplication operations. According to the mathematical statistics and probability theory, the total number of random selections is $C_m^{m'} = \frac{m(m-1)(m-2)\cdots(m-m'+1)}{m'!}$. Then, system smart device matches the multiplication of index $\{I_{w_j}\}_{j\in[1,m]}$ with request (T_1, T_2). As long as there is one successful match in the $C_m^{m'}$ times matching, it proves that the equation holds. that is ti say that the DU has right to access the medical record containing the keywords $w' = \{w'_1, w'_2, \cdots, w'_m\}$. Then system smart device returns a search token $ST_w = (ID, h, \gamma)$ to DU, in which contains a block ID, a hash address h and a random number γ. Otherwise, returns $\perp$.

Verify1: DU can get the tuple (h_R, h') from the blockchain according to the block ID which is contained in the search token ST_w. Then he computes $h'' = h^\gamma$ and compares it with h' to verify the correctness of h. If $h'' = h'$, he leverages the hash address h to download medical records containing the keyword $w' = \{w'_1, w'_2, \cdots, w'_m\}$ from the IPFS.

Decrypt: After DU gets the ciphertext

$$CT = \left\{C', C_0, \left\{C_{i,1}, C_{i,t,2} | 1 \leqslant t \leqslant m_i, 1 \leqslant i \leqslant m\right\}\right\},$$

he decrypts the ciphertext with his secure private key sk_S, for each attribute number $\{i|1 \leqslant i \leqslant m\}$, $C'_{i,2} = C_{i,t_i,2}$ where $S_i = v_{i,t_i}$, then

$$R' = \frac{C' \cdot \prod_{i=1}^{m} e\left(C_{i,1}, d_{i,1}\right)}{e\left(C_0, d_0\right) \cdot \prod_{i=1}^{m} e\left(C_{i,2'}, d_{i,2}\right)}.$$

Verify2: The DU hashes medical record R' with SHA256 hash function and compares it with the hash value on the blockchain. If $h'_R = h_R$, the medical record R' is original and has not been tampered with.

VI. SECURITY ANALYSES

A. CORRECTNESS

During the search process, if the attributes set $S = \{S_1, S_2, \cdots, S_m\}$ satisfies the access policy $W = \{W_1, W_2, \cdots, W_m\}$, where $W_i \in L_i$, then,

$$\begin{aligned} e(I_2, T_1) &= e\left(\left(u_1 \cdot \prod_{\forall i,t\in W_i} A_{i,t}\right)^y, D_1^{\overline{H(w'_1\|w'_2\|\cdots\|w'_m)}^{z}}\right) \\ &= e\left(\left(g_1^\sigma \cdot \prod_{\forall i,t\in W_i} g_1^{a_{i,t}}\right)^y, g_2^{s\cdot\overline{H(w'_1\|w'_2\|\cdots\|w'_m)}^{z}}\right) \\ &= e(g_1, g_2)^{\overline{H(w'_1\|w'_2\|\cdots\|w'_m)}^{\sigma ysz}} \cdot e(g_1, g_2)^{\overline{H(w'_1\|w'_2\|\cdots\|w'_m)}^{ysz \sum_{\forall i,t\in W_i} a_{i,t}}} \\ e(I_1, T_2) &= e\left(g_1^{\overline{H(w_1\|w_2\|\cdots\|w_m)}^{y}}, D_2^z\right) \\ &= e\left(g_1^{\overline{H(w_1\|w_2\|\cdots\|w_m)}^{y}}, \left(g_2^\sigma \cdot g_2^{\sum_{\forall i,t\in W_i} a_{i,t}}\right)^{sz}\right) \\ &= e(g_1, g_2)^{\overline{H(w_1\|w_2\|\cdots\|w_m)}^{y\sigma sz}} \cdot e(g_1, g_2)^{\overline{H(w_1\|w_2\|\cdots\|w_m)}^{ysz \sum_{\forall i,t\in W_i} a_{i,t}}} \end{aligned}$$

If $w' = w$, and $e(I_1, T_2) = e(I_2, T_1)$, this means that the DU has privilege to access the medical record containing the keywords $w' = \{w'_1, w'_2, \cdots, w'_m\}$.

During the decryption process, if attributes set $S = \{S_1, S_2, \cdots, S_m\}$ satisfies the access policy $W = \{W_1, W_2, \cdots, W_m\}$, where $W_i \in L_i$, then,

$$R' = \frac{C' \cdot \prod_{i=1}^{m} e\left(C_{i,1}, d_{i,1}\right)}{e\left(C_0, d_0\right) \cdot \prod_{i=1}^{m} e\left(C'_{i,2}, d_{i,2}\right)} = \frac{R \cdot e\left(g_1, g_2\right)^{\alpha r} \cdot \prod_{i=1}^{m} e\left(g_1^{r_i}, g_2^{x_i + a_{i,t_i}\lambda_i}\right)}{e\left(g_1^{\beta r}, g_2^{\frac{\alpha+x}{\beta}}\right) \cdot \prod_{i=1}^{m} e\left(g_1^{a_{i,t}r_i}, g_2^{\lambda_i}\right)} = \frac{R \cdot e\left(g_1, g_2\right)^{\alpha r} \cdot e\left(g_1, g_2\right)^{rx + ra_{i,t}\lambda_i}}{e\left(g_1, g_2\right)^{r\alpha + rx} \cdot e\left(g_1, g_2\right)^{a_{i,t}r\lambda_i}} = R$$

B. SECURITY PROOF

1) SECURITY PROOF OF ENCRYPTED DATA

At the beginning of the game, the adversary promises two challenge policies W_0, W_1. We use the notation $W_b = \left(W_{b,1}, W_{b,2}, \cdots, W_{b,m}\right)$ to express them, where $b \in \{0, 1\}$. According to the proof in the literature [26], no adversary can win the original game G. Therefore, we prove the security of our scheme by leveraging the indistinguishability of following games $G, G_0, \cdots, G_{l-1}, G_l$.

First, We get a new game G_0 by changing the way of ciphertext component in game G, In game G_0, if the adversary did not obtain the sk_S generated from their attribute S such that $S \models W_0 \wedge S \models W_1$, then no matter what the b is, the adversary chosen a random number in G_T as the ciphtertext component C', and the rest of the components are unchanged. If the adversary obtain the sk_S generated from their attribute S satisfying $S \models W_0 \wedge S \models W_1$, then no matter what the b is, the adversary generates component C' is same as in game G. That is to say, $G = G_0$ in this case.

Theorem 1: In the setting of the security parameter k, the difference for the advantage of adversary A between game G and game G_0 is negligible according to the assumption of asymmetric DBDH.

Proof: The simulator given an asymmetric DBDH challenge tuple

$$\left[g_1, g_2, g_1^{Z_1}, g_1^{Z_2}, g_2^{Z_2}, g_2^{Z_3}, Z\right],$$

where $Z = e\left(g_1, g_2\right)^{Z_1 Z_2 Z_3}$ or a random number with same probability.

Init: The adversary A promises two challenge ciphertext policies W_0 and W_1 to simulator B, then B flips a random coin b, in which $b \in \{0, 1\}$. If $b = 1$, then $Z = e(g_1, g_2)^{z_1 z_2 z_3}$, else Z is a random element.

Setup: The simulator B first inputs a security parameter λ and gets the system public parameter. Then he runs KeyGen algorithm and generates the public key pk_o for adversary A. The simulator B first sets $Y = e\left(g_1, g_2\right)^{z_1 z_2}$, $B = g_1^{z_2}$, $u_1 = g_1^{z_3}$, this implies $\alpha = z_1 z_2$, $\beta = z_2$, $\sigma = z_3$. For each attribute number $\{i | 1 \leqslant i \leqslant m\}$, the simulator B generates $\left\{A_{i,t} | 1 \leqslant t \leqslant m_i\right\}$, if $v_{i,t} \in W_{b,i}$, the simulator B sets $A_{i,t} = g_1^{a_{i,t}}$. Otherwise,the simulator B sets $A_{i,t} = g_1^{z_1\, a_{i,t}}$, where $\left\{a_{i,t} \in Z_p^* | 1 \leqslant t \leqslant m_i\right\}$ is selected randomly. Finally the simulator B publishes public parameters $\left\{Y, B, u_1, \left\{A_{i,t} | 1 \leqslant t \leqslant m_i, 1 \leqslant i \leqslant m\right\}\right\}$.

Phase1: Adversary A promises an attribute set $S = [S_1, S_2, \cdots, S_m]$ in a secret key query stage. In our definition, if $S \nvDash W_0 \wedge S \nvDash W_1$, then R_0 is equal with R_1, under this situation, the game G and game G_0 are the equal, So the difference about the advantage of adversary A between game G and in game G_0 is equal. At this point simulator B ends the game and makes a guess randomly. Therefor, we only consider the case: $S \nvDash W_0 \wedge S \nvDash W_1$. When $S \nvDash W_0 \wedge S \nvDash W_1$, there certainly exist $y \in \{1, 2, \cdots, m\}$ such that $S_y = v_{y,t_y} \notin W_{b,y}$. For each attribute number $\{i | 1 \leqslant i \leqslant m\}$, simulator B selects $\lambda_i, x'_i \in Z_p^*$ at random, then sets $x = x' - z_1 z_2$ and $x = \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} x'_i - z_1 z_2$, so the component of the secret key can be computed as

$$d_0 = g_2^{\frac{\alpha+x}{\beta}} = g_2^{\frac{z_1 z_2 + \sum_{i=1}^{m} x'_i - z_1 z_2}{z_2}} = g_2^{\frac{\sum_{i=1}^{m} x'_i}{z_2}}$$
$$d_{y,1} = g_2^{y_k + a_{y,t_y}\lambda_y} = g_2^{x'_y + z_1 z_2 + a_{y,t_y}\lambda_y} = g_2^{x'_y + a_{y,t_y}\lambda'_y},$$

where

$$\lambda_y = \lambda'_y + \frac{Z_1 Z_2}{a_{y,t_y}};$$
$$d_{y,2} = g_2^{\lambda_y}.$$

When $i \neq y$, it sets $x = x'$, simulator B also can compute $\left\{d_0, d_{i,1}, d_{i,2}\right\}$ without any difficult.

challenge: Adversary A promises two challenge records R_0 and R_1. Simulator B flips a coin b, $b \in \{0, 1\}$, and sets $C' = R_b \cdot Z^r$, $C_0 = B^r = g_1^{z_2 r}$. For W_b, if $v_{i,t} \in W_{b,i}$, then simulator B generates $\left\{C_{i,1}, C_{i,t,2}\right\} = \left\{g_1^{r_i}, A_{i,t}^{r_i}\right\}$ correctly because $A_{i,t}$ does not contain unknown z_1, else if $v_{i,t} \notin W_{b,t}$, then he can choose a random number as the components $C_{i,1}, C_{i,t,2}$.

Phase 2: Similar as Phase 1.

Guess: Adversary A makes a guess b'. If $b' = b$, adversary A is running in game G, $Z = e\left(g_1, g_2\right)^{z_1 z_2 z_3}$ and otherwise A is running in game G_0, Z is random. Therefore, through our assumption, the simulator B can win the asymmetric DBDH game with the advantage ε. □

Next, we continue to change the way of ciphertext components generated in game G_0 and to get a new game. For each $v_{i,t}$ in the game G_{l-1}, when $v_{i,t} \in W_{0,t} \wedge v_{i,t} \in W_{1,t}$ or $v_{i,t} \notin W_{0,t} \wedge v_{i,t} \notin W_{1,t}$, we generate ciphertext components $\{C_{i,1}, C_{i,t,2}\}$ normally, and when $v_{i,t} \in W_{0,t} \wedge v_{i,t} \notin W_{1,t}$ or $v_{i,t} \notin W_{0,t} \wedge v_{i,t} \in W_{1,t}$, no matter what value the random coin b is, we choose the random number in the game G_l to replace the ciphertext components $\{C_{i,1}, C_{i,t,2}\}$ generated normally in the game G_{l-1}. We replace the ciphertext component normally generated in the previous game with the random value

from the next game until there is no $v_{i,t} \in W_{0,t} \wedge v_{i,t} \notin W_{1,t}$ or $v_{i,t} \notin W_{0,t} \wedge v_{i,t} \in W_{1,t}$. Every time we replace it, a new game is formed. In the last game, due to the ciphertext components are in the same distribution, so the adversary's advantage is 0. And then we can embed the D-Linear challenge in this way, so the distinction of the game G_l and the game G_{l-1} is called D-Linear challenge.

Theorem 2: In the setting of the security parameters k. The difference about the advantages of adversary A between game G_{l-1} and the game G_l is negligible according to the D-Linear assumption.

Proof: The challenger given a D-Linear challenge tuple

$$\left[g_1, g_2, g_1^{z_1}, g_1^{z_2}, g_2^{z_1}, g_1^{z_1 z_3}, Z, g_2^{z_1 z_3}, g_1^{z_3+z_4}\right],$$

where $Z = g_1^{z_2 z_4}$ or Z is a random number with same probability. we might as well assume that $v_{i,t} \in W_{0,t} \wedge v_{i,t} \notin W_{1,t}$.

Init: The adversary A promises two challenge ciphertext policies W_0 and W_1 to simulator B, then B flips a coin b randomly, $b \in \{0, 1\}$. If $b = 0$, simulator B ends the challenge and makes a guess randomly, Clearly, in our definition, formula $b = 0$ means $v_{i_l,t_l} \in W_{1,i_l} \wedge v_{i_l,t_l} \notin W_{0,i_l}$, then it obvious that $G_{l-1} = G_l$. Because the distribution of ciphertext in game G_{l-1} and in game G_l are same, so there is no advantage for adversary A between the game G_{l-1} and game G_l. Thus, we need to assume $b = 1$.

Setup: The simulator B first inputs a security parameter λ and gets the system public parameter, then he runs KeyGen algorithm and generates the public key pk_o for adversary A. The simulator B first sets $Y = e(g_1, g_2)^{z_1}$, $B = g_1^{z_3}$, $u_1 = g_1^{z_2 z_3}$, this implies $\alpha = z_1$, $\beta = z_3$, $\sigma = z_2 z_3$. For each attribute number $\{i | 1 \leqslant i \leqslant m\}$, $\left\{A_{i,t} | 1 \leqslant t \leqslant m_i\right\}$ is generated by the simulator B. If $v_{i,t} \in W_{b,i}$, the simulator B sets $A_{i,t} = g_1^{\alpha_{i,t}}$; Else, he sets $A_{i,t} = g_1^{z_1 \alpha_{i,t}}$, where $\alpha_{i,t} \in Z_p^* | 1 \leqslant t \leqslant m_i$ is random. Finally the simulator B publishes public parameters as $\left\{Y, B, u_1, \left\{A_{i,t} | 1 \leqslant t \leqslant n_i, 1 \leqslant i \leqslant m\right\}\right\}$.

Phase 1: Adversary A promises an attribute set $S = [S_1, S_2, \cdots, S_m]$ in a secret key query stage. If $S_{i_l,t_l} \notin v_{i_l,t_l}$, simulator B can compute the secret key $\left\{d_{i_l,1}, d_{i_l,2}\right\}$ without any difficult. Then we assume $S_{i_l,t_l} = v_{i_l,t_l}$, and simulator B computes the components $\left\{d_{i_l,1}, d_{i_l,2}\right\}$ as follows:

$$d_{i_l,1} = g_2^{x_{i_l} + a_{i_l,t_l} \lambda_{i_l}} = g_2^{x'_{i_l}},$$

where

$$x_{i_l} = x'_{i_l} - a_{i_l,t_l} \lambda_{i_l}$$
$$d_{i_l,2} = g_2^{\lambda_{i_l}}.$$

Now, we assume $S \nvDash W_0 \wedge S \nvDash W_1$, because $S_{i_l} = v_{i_l,t_l} \notin W_{1-b,i_l}$, that is to say $S \nvDash W_{1-b}$, therefore $S \nvDash W_b$, so there certainly exist $y \in \{1, 2, \cdots, m\}$ satisfying $S_y = v_{y,t_y} \notin W_{b,y}$. Then B sets $x_y = x'_y + a_{i_l,t_l} \lambda_{i_l}$ and computes the components $\left\{d_{y,1}, d_{y,2}\right\}$ as follows:

$$d_{y,1} = g_2^{x_y + a_{y,t_y} \lambda_y} = g_2^{x'_y + a_{i_l,t_l} \lambda_{i_l} + a_{y,t_y} \lambda_y} = g_2^{x'_y + a_{y,t_y} \lambda'_y},$$

where

$$\lambda_y = \lambda'_y - \frac{a_{i_l,t_l} \lambda_{i_l}}{a_{y,t_y}};$$
$$d_{y,2} = g_2^{\lambda_y}.$$

Also, when $i \neq i_l$ or y, simulator B can also compute $\left\{d_{i,1}, d_{i,2}\right\}$ easily. Finally

$$\begin{aligned} x &= \sum_{i=1}^{m} x_i = x_{i_l} + x_y + \sum_{i \neq i_l, y}^{m} x_i \\ &= x'_{i_l} - a_{i_l,t_l} \lambda_{i_l} + x'_y + \sum_{i \neq i_l, y}^{m} x_i \\ &= x'_{i_l} + x'_y + \sum_{i \neq i_l, y}^{m} x_i \end{aligned}$$

he can compute the component $d_0 = \dfrac{\alpha + x}{\beta}$ with ease.

Challenge: Adversary A promises two challenge records R_0 and R_1, simulator B sets $C_0 = g_1^{\beta r} = g_1^{z_2(z_3+z_4)}$, this implies $r = z_3 + z_4$. If S satisfies $S \nvDash W_0 \wedge S \nvDash W_1$ in each query, B choose a random number as C', otherwise B lets $C' = R_b \cdot e(g_1, g_2)^{z_1(z_3+z_4)}$. For W_b, the ciphertext components $\left\{C_{i,1}, C_{i,t,2} | 1 \leqslant i \leqslant m, 1 \leqslant t \leqslant m_i\right\}$ are generated as game G_{l-1}, additional components $C_{i_l,1}, C_{i_l,t_l,2}$ are computed as

$$C_{i_l,1} = g_1^{r_{i_l}} = g_1^{z_4}$$
$$C_{i_l,t_l,2} = A_{i_l,t_l}^{r_{i_l}} = \left(g_1^{\alpha_{i_l,t_l}}\right)^{r_{i_l}} = Z$$

It is implies $r_{i_l} = z_4$, $\alpha_{i_l,t_l} r_{i_l} = z_2 z_4$ and $Z = g_1^{z_2 z_4}$. If $Z = g_1^{z_2 z_4}$, the ciphertext components are generated properly and adversary A just runs game G_{l-1}.

Phase 2: Similar as Phase 1.

Guess: Adversary A makes a guess b'. If $b' = b$, adversary A runs game G_{l-1}, $Z = g_1^{z_2 z_4}$ and otherwise A runs game G_l, Z is random. Therefore, through our assumptions, the advantage of the simulator B is ε during running the D-Linear game. □

2) CHOSEN KEYWORDS ATTACK (CKA) SECURITY

Theorem 3: Under the asymmetric DDH1v1 assumption, it is negligible that the advantage of any polynomial adversary A to break the CKA game, then challenger can construct a simulator B such that B can win the DDH1v1 game with the advantage of $\dfrac{\varepsilon}{2}\left(1 - \dfrac{N^2}{P}\right)$, where $N = \sum_{i=1}^{m} m_i$ [27].

Proof: Simulator B is given an asymmetric DDH1v1 tuple $\left(g_1, g_2, g_1^{z_1}, g_1^{\frac{z_2}{z_1}}, Z\right)$.

Init: The simulator B gives a security parameter λ and gets the system public parameter, and then B runs KeyGen algorithm. The adversary A promises an access policy $W^* = \left[W_1^*, W_2^*, \cdots, W_m^*\right]$ to simulator B.

TABLE 2. Function comparison.

scheme	Decentralizing	Keyword search	Verifiable search results	Search privacy
[28]	√	√	×	√
[29]	√	√	×	√
[30]	×	√	×	√
[14]	×	√	√	√
ours	√	√	√	√

TABLE 3. Storage cost.

Algorithm	VKSE	ABKS-UR	Ours
Setup\ Init	$6\|G\|+\|G_0\|+\|Z_p\|$	$(3\|S\|+1)(\|G\|+\|Z_p\|)+\|G_0\|$	$\|G_1\|+\|G_2\|+(3+\|S\|)\|Z_p\|$
KeyGen	$(2\|S\|+3)\|G\|+\|Z_p\|$	$(2\|S\|+1)\|G\|+\|Z_p\|$	$\|G_T\|+3\|G_1\|+(2+3\|S\|)\|G_2\|+(3+\|S\|)\|Z_p\|$
Trap\Index	$(2\|S\|+3)\|G\|$	$(2\|S\|+1)\|G\|+\|Z_p\|$	$(1+\|S\|)\|G_1\|$
Search	$(3\|S\|+1)\|G_0\|$	$(\|S\|+2)\|G_0\|$	$(1+\|S\|)\|G_2\|+\|G_T\|$
Verify	$\varphi\|Z_p\|+\|G\|+\|G_0\|$	–	$\|Z_p\|$
Dec	$\varphi\|Z_p\|$	–	–

Note: Symbol φ represents the number of search results.

Setup: Simulator B selects $a'_{i,t} \in Z_P^*$ at random, where $1 \leqslant i \leqslant m$, $1 \leqslant t \leqslant m_i$, and computes

$$A_{i,t} = g_1^{a_{i,t}} = \begin{cases} g_1^{a'_{i,t}}, & v_{i,t} \in W^* \\ \left(g_1^{z_2}\right)^{a'_{i,t}}, & v_{i,t} \notin W^* \end{cases}$$

Then simulator B publishes public parameter as

$$\left\{g_1, g_2, \alpha, \beta, \sigma, \left\{a_{i,t} | 1 \leqslant i \leqslant n, 1 \leqslant t \leqslant m_i\right\}\right\}.$$

Phase 1: The adversary A promises an attribute set $S = [S_1, S_2, \cdots, S_m]$ to simulator B. B computes

$$\begin{aligned} \sum_{\forall i,t \in S} a_{i,t} &= \sum_{\forall i,t \in W^*} a_{i,t} + \sum_{\forall i,t \notin W^*} a_{i,t} \\ &= \sum_{\forall i,t \in W^*} a'_{i,t} + b \sum_{\forall i,t \notin W^*} a_{i,t} \\ &= A_1 + bA_2 \end{aligned}$$

next he selects $s \in Z_P^*$ randomly and computes $D_1 = g_2^s$, $D_2 = \left(u_2 g_2^{\sum_{\forall i,t \in S} a_{i,t}}\right)^s = \left(g_2^{\sigma} g_2^{A_1} \left(g_2^b\right)^{A_2}\right)^s$, finally he sends (D_1, D_2) to adversary A. Adversary A generates an access request with this search private key during the search phase.

Challenge: Adversary A submits two keywords w_1, w_2, the simulator B encrypts w_b by flipping a coin b at random. Note that if $b = 0$, then $w_b = \left\{w_{0,1}, w_{0,2}, \cdots, w_{0,m}\right\}$, and if $b = 1$, then $w_b = \left\{w_{1,1}, w_{1,2}, \cdots, w_{1,m}\right\}$. Simulator B sets $y = z_1 z_2$, then he computes $T_1 = Z^{\frac{1}{H\left(w_{b,1} \| w_{b,2} \| \cdots \| w_{b,m}\right)}}$ and $T_2 = \left(u_1 \prod_{\forall i,t \in W^*} A_{i,t}\right)^y = Z^{\sigma + \sum_{\forall i,t \in W^*} a'_{i,t}}$ simultaneously. Then he sends the request (T_1, T_2) to adversary A.

Phase 2: Similar as Phase 1.

Guess: Adversary A makes a guess b', if $b' = b$, B responds DDH and otherwise he responds random.

If $Z = g_2^{z_1 z_2}$, I is a correct index, and the advantage of A is ε, thus

$$\begin{aligned} &Adv\left[B \rightarrow \text{“}DDH\text{”} | Z = g_2^{z_1 z_2}\right] \\ &= Adv\left[b' = b | Z = g_2^{z_1 z_2}\right] \\ &= \frac{1}{2} + \varepsilon \end{aligned}$$

If $Z = g_2^z$, I is not a correct index, the advantage of A is 0, thus

$$\begin{aligned} &Adv\left[B \rightarrow \text{“}random\text{”} | Z = g_2^z\right] \\ &= Adv\left[b' = b | Z = g_2^z\right] \\ &= \frac{1}{2} \end{aligned}$$

According to the above information in the DDH game, the advantage of B is $\frac{\varepsilon}{2}\left(1 - \frac{N^2}{P}\right)$ under the DDH game.

VII. PERFORMANCE ANALYSIS

A. FUNCTION COMPARISON

We compared some existing schemes with our scheme in decentralizing, keyword search, verifiable search results, and search privacy, as shown in Table 2. The symbol "√" indicates that the scheme has this function, and "×" indicates the opposite case. As can be seen from the table, scheme [14] and scheme [30] dont meet the feature of decentralizing, and scheme [14] supports the search results verifiable; The schemes [28] and [29] cannot verify search results; our scheme is more powerful and can support the above features at the same time. These features make our scheme more suitable for practical applications.

B. STORAGE COST COMPARISON

In order to compare the storage costs in the three schemes, we first define some symbols, we set $|G|$, $|G_0|$, $|G_T|$ to represent the bit length of an element in group G, G_0, and G_T, respectively. Meanwhile, we use symbol$|Z_p|$ to denote the bit length of an element in filed Z_p, symbol $|S|$ to represent the number of the DU's attribute. The storage cost of the three scheme as shown in Table 3.

From Table 3 we can know that although our scheme has higher storage cost in KeyGen algorithm than scheme [14] and scheme [11], the storage cost in Setup/Init, Trap/Index and Search is all significantly lower than them.

In Alice's example, according to the experiment simulation in our scheme, we set $|\ Z_p\ | = 160\ bit$,$|\ G_1\ | = |\ G_2\ | = |\ G_T\ | = 1024\ bit$, thus the storage cost in Setup algorithm, KeyGen algorithm, Index algorithm, Search algorithm, and Verify

TABLE 4. Computational cost.

Algorithm	VKSE	ABKS-UR	Ours
Setup\ Init	$5O_E + O_p + O_{E_0}$	$3\|S\|O_E + O_{E_0} + O_p$	0
KeyGen	$(2\|S\| + 4)\, O_E$	$(2\|S\| + 1)\, O_E + 2O_{E_0}$	$O_p + (2 + \|S\|)\, O_{E_1} + (3 + 3\|S\|)\, O_{E_2}$
Trap\Index	$(2\|S\| + 4)\, O_E$	$(2\|S\| + 1)\, O_E$	$(1 + \|S\|)\, O_{E_1}$
Search	$(1 + 2\|S\|)\, O_p + \|S\|O_{E_0}$	$(\|S\| + 1)\, O_p + O_{E_0}$	$(2 + \|S\|)\, O_{E_2}$
Verify	$(2\varphi + 1)\, O_E + 2O_p$	–	O_{E_1}
Dec	$O_p + O_{E_0}$	–	$(2 + 2\|S\|)\, O_p$

Note: Symbol φ represents the number of search results.

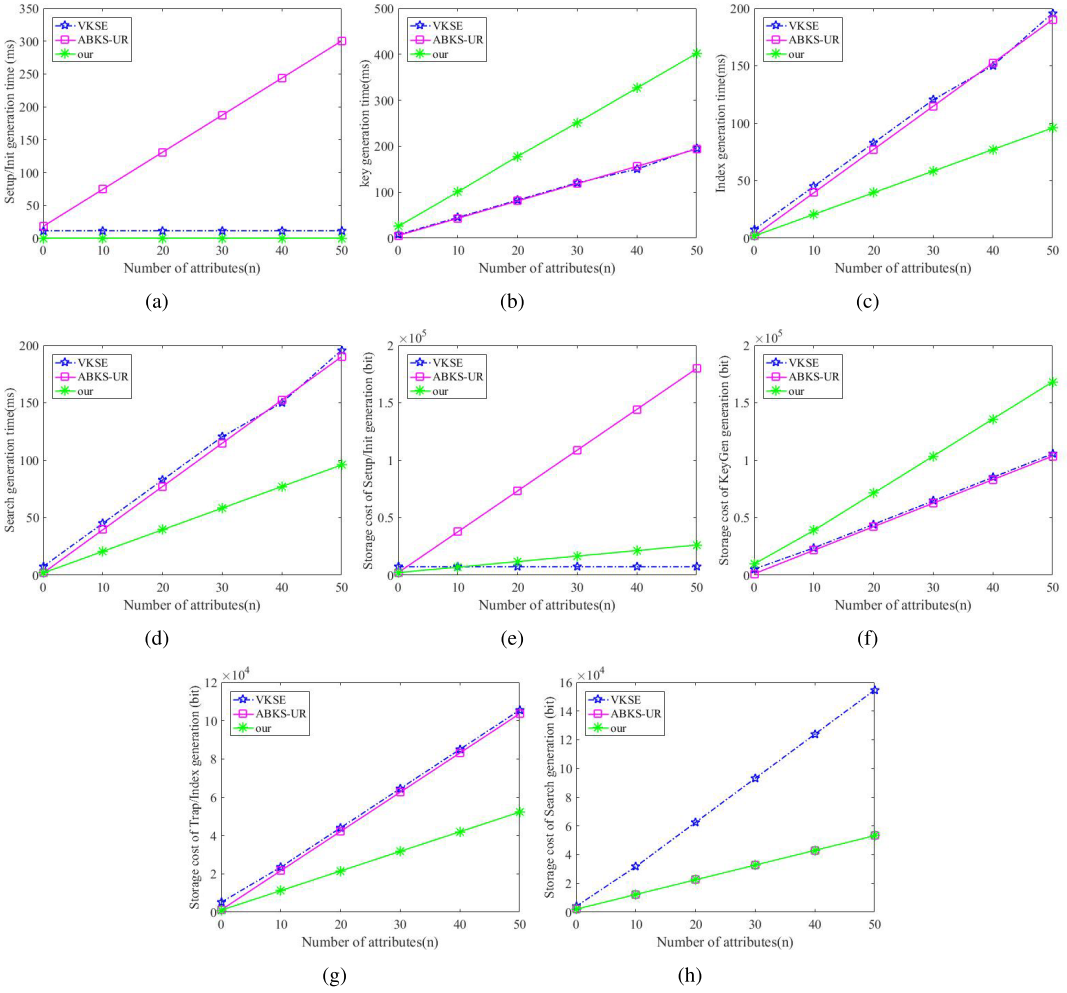

FIGURE 4. The storage and computational costs analysis obout different algorithm in our scheme, VKSE scheme and ABKS-UR scheme respectively.(a)Setup generations time.(b)KeyGen generations time.(c)Index generations time.(d)Search generations time.(e)Storage cost of Setup. (f)Storage cost of KegGen.(g)Storage cost of Index.(h)Storage cost of Search.

algorithm is 3008 *bit*, 21504 *bit*, 4096 *bit*, 5120 *bit* and 160 *bit*, respectively. So, the storage cost of the example is 33888*bit*.

C. COMPUTATIONAL COST COMPARISON

To analyze the computational costs of the three schemes, we first define some time-consuming operation symbols, namely bilinear pairing operation O_P, exponentiation operations O_E (exponential operation O_{E_0} in group G_0, exponential operation O_{E_1} in group G_1 and exponential operation O_{E_2} in group G_2).

From Table 4, it obvious that our scheme is most efficient in the Init, Index and Search algorithms except the KeyGen algorithm. In addition, during that the Verify algorithm based the different result verification mechanism, so we analyzed the computational cost just in our scheme and the VKSE scheme.

From the above information, we can conclude that our scheme enables access control without increasing additional

storage and computational overhead. However, this fact needs further verification.

D. EXPERIMENTAL SIMULATION

We leverage the Paring Based Cryptography (PBC) laboratory to evaluate the actual performance of the three solutions, and our experiments were performed on a Ubuntu server 15.4 with a processing core i5 processor using real data set. In our experiments, we set $|Z_p| = 160\ bit$, $|G_1| = |G_2| = |G_T| = 1024\ bit$, respectively. Due to our scheme, VKSE scheme, and ABKS-UR scheme are all affected by attributes. We let $|S| \in [0, 50]$ in KeyGen, Index and Search algorithms to show the degree which it is affected by the number of attributes.

Because we need separately generate the search private key and secret private key in the Setup algorithm which associate with their attribute, we can see from Fig 4 (b) and (f) that the storage and computational overheads in our scheme is a little bit higher than VKSE and ABKS-UR scheme. However, such a design let us no longer consider the factor of attribute in Index and Search algorithm, and thus the efficiency in our scheme be improved availably. In fact, we can see clearly in other figure, the storage and computational overheads in our scheme all lower than VKSE and ABKS-UR scheme. So our scheme is security and efficient as a whole.

To sum up, the actual analysis using the real data set is consistent with the theoretical analysis. In our scheme, almost all the storage and computational costs are lower than VKSE and ABKS-UR scheme. Also, our scheme gains access control without adding additional storage and computational cost, at the same time, it provides a verification mechanism for the results. Therefore, the experimental simulations show that our scheme is security and efficient.

VIII. CONCLUSION

We propose a new encryption scheme for secure storage and efficient sharing of electronic medical records based on the attribute-based encryption system, blockchain technology, and the InterPlanetary File System (IPFS) storage platform. The scheme provides good access control for the electronic medical records using attribute-based encryption technology so that people who are not related to the patient cannot see the private data of the patient without authorized. At the same time, the IPFS storage platform was introduced to replace the semi-honest but curious cloud server to ensure the secure storage of the medical data, and the blockchain technology records the storage and search process, ensuring the originality and traceability of the data and solving the security deficiency of central authority. However, our scheme still has some shortcomings, such as the access privilege and the timeliness of expired user, then the functional issues of the data stored in the blockchain. In the following work, we will consider using attribute revocation function to solve the problem of access privilege of expired users. At the same time, the smart contract will be embedded in our scheme to make the data on the blockchain more timeliness and functional.

REFERENCES

[1] S. Mhatre and A. V. Nimkar, "Secure cloud-based federation for EHR using multi-authority ABE," in *Advanced Computing and Intelligent Engineering*. Berlin, Germany: Springer, 2019, pp. 3–15.
[2] R. Gandikota and B. E. Reddy, "Secure architecture to manage EHR's in cloud using SSE and ABE," *Health Technol.*, vol. 5, no. 3, pp. 195–205, 2015, doi: 10.1007/s12553-015-0116-0.
[3] B. E. Reedy and G. Ramu, "A secure framework for ensuring EHR's integrity using fine-grained auditing and CP-ABE," in *Proc. IEEE IEEE 2nd Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput. (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, New York, NY, USA, Apr. 2016, pp. 85–89.
[4] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-Health systems via consortium blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 140, Aug. 2018.
[5] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, and S. Liu, "Blockchain-based data preservation system for medical data," *J. Med. Syst.*, vol. 42, no. 8, p. 141, Aug. 2018.
[6] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K.-R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan. 2018.
[7] H. Wang and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 152, Aug. 2018.
[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT*, R. Cramer, Ed. Berlin, Germany: Springer, 2005, pp. 457–473.
[9] F. Qi, Y. Li, and Z. Tang, "Revocable and traceable key-policy attribute-based encryption scheme," *J. Commun.*, vol. 39, pp. 63–69, Nov. 2018.
[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Berkeley, CA, USA, May 2007, pp. 321–334.
[11] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr. 2014, pp. 226–234.
[12] C. Guo, R. Zhuang, Y. Jie, Y. Ren, T. Wu, and K.-K.-R. Choo, "Fine-grained database field search using attribute-based encryption for E-healthcare clouds," *J. Med. Syst.*, vol. 40, no. 11, p. 235, Nov. 2016.
[13] H. Su, Z. Zhu, L. Sun, and N. Pan, "Practical searchable CP-ABE in cloud storage," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Chengdu, China, Oct. 2016, pp. 180–185.
[14] Y. Miao, J. Ma, Q. Jiang, X. Li, and A. K. Sangaiah, "Verifiable keyword search over encrypted cloud data in smart city," *Comput. Electr. Eng.*, vol. 65, pp. 90–101, Jan. 2018.
[15] W. M. McKeeman, "Representation error for real numbers in binary computer arithmetic," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 5, pp. 682–683, Oct. 1967.
[16] S. E. Wang and B. G. Lin, "A scheme of attribute-based encryption access policy used in mobile cloud storage for personal health records," in *Proc. Int. Conf. Inf. Netw. Secur. (ICINS)*, Beijing, China, 2014, pp. 193–200.
[17] S. Alshehri, S. P. Radziszowski, and R. K. Raj, "Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption," in *Proc. IEEE 28th Int. Conf. Data Eng. Workshops*, Arlington, VA, USA, Apr. 2012, pp. 143–146.
[18] L. Xu, C. Xu, J. K. Liu, C. Zuo, and P. Zhang, "Building a dynamic searchable encrypted medical database for multi-client," *Inf. Sci.*, early access, May 24, 2019, doi: 10.1016/j.ins.2019.05.056.
[19] Q. Zheng, Y. Li, P. Chen, and X. Dong, "An innovative IPFS-based storage model for blockchain," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, vol. 1, Dec. 2018, pp. 704–708.
[20] Y. Chen, H. Li, and K. Li, "An improved P2P file system scheme based on IPFS and Blockchain," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Boston, MA, USA, Dec. 2017, pp. 2652–2657.
[21] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied Cryptography and Network Security*. Springer, 2008, pp. 111–129.
[22] V. Goyal, A. Jain, A. Sahai, and O. Pandey, "Bounded ciphertext policy attribute based encryption," in *Proc. Int. Colloq. Automata Lang. Program. (ICALP)*. Springer, 2008, pp. 579–591.
[23] H. Wang and Y. Peng, "A CP-ABE access control scheme based on proxy re-encryption in cloud storage," in *Proc. Int. Conf. Cloud Comput. Secur. (ICCCS)*. Springer, 2018, pp. 413–425.

[24] Y. S. Rao and R. Dutta, "Computational friendly attribute-based encryptions with short ciphertext," *Theor. Comput. Sci.*, vol. 668, pp. 1–26, Mar. 2017.
[25] H. Su, Z. Zhu, L. Sun, and N. Pan, "Practical searchable CP-ABE in cloud storage," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Chengdu, China, Oct. 2016, pp. 180–185.
[26] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Berkeley, CA, USA, May 2007, pp. 321–334.
[27] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VI, USA, 2006, pp. 89–98.
[28] P. Jiang, F. Guo, K. Liang, J. Lai, and Q. Wen, "Searchain: Blockchain-based private keyword search in decentralized storage," *Future Gener. Comput. Syst.*, vol. 107, pp. 781–792, Jun. 2020.
[29] H. G. Do and W. K. Ng, "Blockchain-based system for secure data storage with private keyword search," presented at the IEEE World Congr. Services (SERVICES), Jun. 2017, doi: 10.1109/SERVICES.2017.23.
[30] N. H. Sultan, N. Kaaniche, M. Laurent, and F. A. Barbhuiya, "Authorized keyword search over outsourced encrypted data in cloud environment," *IEEE Trans. Cloud Comput.*, early access, Jul. 30, 2019, doi: 10.1109/TCC.2019.2931896.

JIN SUN was born in Anhui, China, in 1977. She received the B.S. degree from Shaanxi Normal University, in 2000, the M.S. degree from the Xi'an University of Technology, in 2005, and the Ph.D. degree from Xidian University, in 2012. She is currently an Associate Professor with the Xi'an University of Technology. Her research interests include cryptography, information security, network security, and blockchain.

XIAOMIN YAO was born in Gansu, China, in 1994. She received the B.S. degree from Tianshui Normal University, in 2018. She is currently pursuing the M.S. degree with the Xi'an University of Technology. Her research interests include cryptography and information security.

SHANGPING WANG was born in Shaanxi, China, in 1963. He received the B.S. degree from the Xi'an University of Technology, in 1982, the M.S. degree from Xi'an Jiaotong University, in 1989, and the Ph.D. degree from Xidian University, in 2003. He is currently a Professor with the Xi'an University of Technology. His research interests include cryptography, information security, blockchain, and the Internet of Things.

YING WU was born in Shandong, China, in 1997. She received the B.S. degree from Qufu Normal University, in 2019. She is currently pursuing the M.S. degree with the Xi'an University of Technology. Her research interests include cryptography, information security, and blockchain.