IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 16, 2020, accepted October 23, 2020, date of publication November 3, 2020, date of current version November 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035468

Anonymous Electronic Health Record Sharing Scheme Based on Decentralized Hierarchical Attribute-Based Encryption in Cloud Environment

XUEYAN LIU[1], XIAOTAO YANG[1], YUKUN LUO[1], LI WANG[1], AND QIANG ZHANG[2]
[1]College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, China
[2]College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

Corresponding author: Xueyan Liu (liuxy@nwnu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61662071 and Grant 71764025.

ABSTRACT The rapid development of communication technologies, the network, advanced computing methods and wireless medical sensors gives rise to a modern medical system. In this system, large-scale electronic health records (EHRs) are often outsourced to be stored at the third parties, such as cloud service providers (CSPs). However, CSPs are not trustworthy, that is, serious security and privacy concerns about cloud service exist because it may expose the user's sensitive data to CSPs or unauthorized users in transmission, storage and sharing. To prevent the privacy disclosure of patients better and realize information sharing more effectively, this paper proposes an anonymous EHRs sharing scheme based on decentralized hierarchical attribute-based encryption (ABE). In the proposed scheme, (1) Multiple attribute authority (AA) ABE is leveraged to achieve fine-grained and scalable data access control and avoid bottleneck. Meanwhile, hierarchical access tree is used to encrypt multiple files in one operation, thereby saving calculation and storage load greatly. Moreover, the hidden access policy enhances user privacy protection. (2) The global identifier (GID) of a user is introduced to resist the collusion attack of users. Subsequently, an anonymous key generation mechanism is equipped to prevent multiple AAs from building a full profile using the user's GID. (3) To ensure the correctness and integrity of EHRs, users can conduct double verification based on the verification tag and convergent key. Finally, the efficiency analysis and experiments show that the scheme meets the security requirements of key management and privacy preservation in cloud and is proven secure and efficient in practice under the decisional bilinear Diffie-Hellman (DBDH) assumption.

INDEX TERMS Electronic medical records, multiple attribute-authority, hierarchical access tree, decentralized, privacy preservation.

I. INTRODUCTION

With the rapid development of science and technology, sensors, data processors and communication monitor devices are widely used in our daily life. Meanwhile, traditional medical models cannot meet the needs of most people. Such as those who need providing real-time and continuous monitoring for the elderly, the young and the disabled and the opportunity to prevent diseases for the sub-health groups unable to go to the hospital regularly and conveniently for examination or treatment, of course, the limitation of medical resources is also an important reason. Therefore, some scholars apply wireless sensor technology, which is a dynamic monitoring technology used for constructing modern medical systems. An example of this technology is the wireless body area network (WBAN). WBAN is popular to the masses, and it relies on various kinds of wearable devices and wireless communication technologies to realize continuous remote monitoring and to record and manage the health parameters of patients with chronic diseases, such as diabetes, asthma and heart disease [1]. The amount of electronic medical data is increasing sharply as the modern medical system gradually enters people's lives. Thus, increasing number of electronic health records (EHRs) are outsourced to the cloud because of the advantages of cloud computing technologies, which reduce the load brought by the storage of EHRs, facilitate

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo.

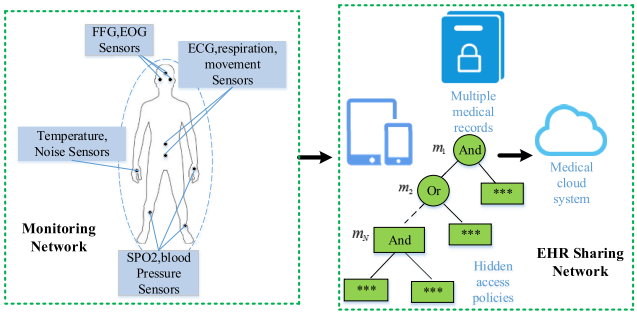

FIGURE 1. A typical modern medical system.

the query of users and improve the efficiency of medical personnel. Figure. 1 shows a typical modern medical system. However, data outsourcing to untrusted third parties has also raised some privacy and security concerns because the EHRs include patients' private information (e.g. name, address and medical history) and can be visited by different users (such as attending doctor, family, care nurses and medical researchers). Therefore, to protect the privacy of patients and prevent unauthorized users from accessing these private data, encrypting EHRs prior to outsourcing is the preferred measure. However, in traditional encryption schemes [2]–[6], access control is often transformed into a complex key escrow problem, which leads to large storage costs. Therefore, the traditional one-to-one encryption technology can no longer meet the user's requirements for fine-grained access control. For example, Bob is a patient who wants to share two EHRs, one is about heart disease which can be accessed if satisfying $\{\text{Attending doctor}\} \vee \{\{cardiovascular\} \wedge \{\text{nurse }\}\} \vee \{\{cardiovascular\} \wedge \{\text{medical researcher}\}\}$, the other one is about infectious diseases which can be accessed by $\{Attendingdoctor\}$. How does he implement this sharing and control access? What can he do to protect privacy and reduce computing costs?

In 2007, Sahai *et al.* [2] put forward the first attribute-based encryption (ABE) scheme, which could achieve one-to-many encryption and could be embedded with access control policies in ciphertext or key to realize flexible access control over sharing data. ABE refers to a public key encryption mechanism that has good application prospect in cloud computing [7]–[11] and is very applicable in the modern medical sharing system. For example, to improve recovery and treatment, a patient not only authorizes doctors to access their own EHRs but also permits nurses to read their EHRs to receive facilitated care. Therefore, flexible access control is an indispensable condition of EHR sharing system and is also a critical condition of cloud computing. In particular, ciphertext policy-ABE (CP-ABE) schemes [12]–[19] are used suitably in EHR sharing systems.

Although the existing ABE schemes [20] brought significant benefits to EHRs sharing system, there are three main hindrances to widely adopting ABE in the system in the near future. Firstly, **bottleneck problem**. For a single-authority ABE schemes, a trusted central authority (CA) is responsible for managing all attributes and distributing secret keys for all users. While the number of users in the system is huge, which lead to a bottleneck. Meanwhile, single authority is also vulnerable to centralized attack, resulting in system paralysis. In order to solve the problem, a decentralized approach is needed. Secondly, **key escrow problem**. In traditional ABE schemes, the single AA is trusted, which is not realistic for authority involves human participation, due to "limited theory" and "opportunistic behavior", credibility will be reduced; moreover, due to its knowledge of user' private keys, users' anonymity cannot be guaranteed and it can decrypt all ciphertexts of users. In multi-authority mechanism, these AAs are semi-honest and untrusted, each of them is responsible for a disjoint subset of the attribute universe. A user, in this architecture, must offer a global identity (GID) to each AA, if multiple malicious authority centers join together to collect user attributes by tracking the user's GID, the user's private will be compromised, so the anonymity and the key escrow problem follow. Thirdly, **access structure disclosure.** In EHRs sharing system, not only EHRs are sensitive, but the access structures include sensitive information. As shown in Figure 1, a patient with heart disease uploads his own EHR encrypted by the defined access policy which will easily be categorized such EHR as a disease of heart. Therefore, the access policy should also be hidden.

In addition to the above main challenges, computational complexity is also a influential factor. In general ABE schemes, if a user shares multiple EHRs or wants to present hierarchical data access, then he needs to define multiple access structures and encrypt these EHRs for many times, thereby not only increasing time consumption but also taking up more cloud medical storage space. Wang *et al.* [21] proposed an ABE scheme based on file hierarchy in cloud computing. The layered access structures are integrated into a single access structure, and then the hierarchical files are encrypted with the integrated access structure. This scheme has low storage cost and computation overhead in terms of encryption and decryption, which technology is suitable for medical system applications to provide multiple files sharing. However, it is a single authority scheme and cannot solve bottleneck problem or key escrow problem.

To sum up, an efficient and practical EHRs sharing system must be distributed and should support anonymous key generation and save storage space.

A. MOTIVATION AND OUR WORK

Over the years, modern medical sharing system has been widely used to provide convenient medical services. Because of its advantages, ABE is introduced in modern medical sharing system. However, when the medical sharing system based on ABE is actually deployed, user privacy, data privacy and security have attracted lots of concern from academicians and industry.

In the system, data visitors (data users) do not want others, including authorities, to know their browsing contents, and data owners (patients) do not want unauthorized users

to know their medical information. In our scheme, anonymous key generation is introduced to improve the anonymity of users and resist the joint attack of multiple authorities. Another solution we have taken is not only to encrypt the EHRs, but also to hide the access structure to avoid data disclosure of the data owner. Because the EHRs uploaded by patients are not unique, and it is possible to upload EHRs at any time on the spot, most of them use portable mobile clients, which have relatively limited computing power.

To summarize, it is necessary to design an electronic medical record sharing scheme based on decentralized hierarchical ABE that supports anonymous key generation.

Ours work aims to above issues and work on them. The main idea is to employ a decentralized hierarchical ABE. As far as I know, this is the first multiple authorities ABE scheme with hierarchical access tree.

Specifically, we firstly use a unique global identifier (GID) to identify different users and resist the collusion of multiple users. Subsequently, problems caused by multi-authority decentralized ABE, wherein multiple authorities can obtain the user's private key information by tracking their global identity, arise. Thus, the user presents attribute authority (AA) with a hash value that contains a secret random number and his real identity GID instead of using his real identity. As such, AAs cannot obtain the user's real identity. Moreover, when the user receives the private key distributed by the AA, he uses his owner secret random number to obtain the real private key.

Next, we adopt a hierarchical access structure. When encrypting multiple documents or hierarchical data access is needed, we can encrypt these documents at once based on the hierarchical access structure. A verification tag is also generated to provide users with convenient integrity verification.

The main contributions of our protocol are listed as follows:

(1) The scheme supports anonymous interaction between the user and the AA to generate private keys for the user, which can avoid the key escrow attack of dishonest attribute authority;

(2) Decentralized multi-authority is introduced to solve the bottleneck and monopoly of centralized authority, and the global identifier GID is introduced to resist collusion attacks;

(3) The use of hierarchical access tree structure to encrypt multiple files at a time not only provides fine-grained access control but also improves the encryption efficiency and greatly saves storage space. In addition, the hidden access policy improves the confidentiality of shared EHRs;

(4) Data users can make a double verification of ciphertext to ensure its correctness and integrity.

The remainder of this paper is organized as follows. Section 2 introduces some preliminary cryptographic backgrounds used in this paper. Section 3 describes the problem formulations, including system and security models. Section 4 describes the concrete construction of our scheme. Section 5 discusses security. Section 6 discusses performance analysis. Finally, Section 7 concludes this paper.

B. RELATED WORK

In 2005, Sahai and Water [22] firstly proposed a public key encryption scheme based on fuzzy identity (IBE), which is the embryonic form of ABE scheme. ABE can be divided into two schemes: attributes encryption of ciphertext policy (CP-ABE) and attribute encryption of key policy (KP-ABE) [23]. In the KP-ABE, the ciphertext is marked with attributes, and the key is related to the access policy. In contrast to CP-ABE [2], the key is marked with attributes, and ciphertext is related to the access policy.

1) MULTIPLE ATTRIBUTE-AUTHORITY ABE

In general ABE scheme, a trusted central authority is needed to distribute and manage user keys, but this leads to key escrow problem, bottleneck problem and centralized attack problem. For example, in 2017, Liu.et.al [20] proposed a secure sharing of personal health records in cloud computing scheme. Although improves ciphertext-based encryption in the EHRs system and improves decryption efficiency in mobile devices, it relies on single-authority ABE scheme. Hence, this can be a bottleneck as this authority may achieve a key escrow attack, due to its knowledge of all user' private. In order to overcome these problems, many multi-authority ABE schemes been proposed [24]–[26]. In 2007, Chase [24] proposed the first multi-authority ABE scheme. In this scheme, there are multiple authorities to distribute private keys and manage attributes. However, a special central authority (CA) is needed. To address this problem, Lin *et al.* [25] presented a multi-authority ABE scheme without a trusted CA. Later, Chase and Chase *et al.* [26] constructed another multi-authority ABE scheme, in which they use a distributed pseudorandom functions to remove the trusted CA. However, it uses a restricted access structure (limited expressiveness) and has the demonstrated identity-leakage. In 2011, Lewko and Waters. [27] also proposed another multi- authority scheme consisting on issuing attributes and their related secret keys from different attribute authorities. However, user gets his secret keys from multiple authorities and it is very hard to resist user collusion attacks in multi-authority ABE scheme. Lewko and Waters gave a solution by introducing a Unique Global Identifier (GID) for each user to prevent collision attacks. Each user has a unique GID, user's secret keys must be tied to the GID. Although the problem of user collusion attack is solved, it compromises user's privacy. Multiple malicious authorities can collaborate to collect user's attributes by tracing user's GID. To solve this issue, Liang *et al.* [28] proposed a privacy-preserving decentralized ABE scheme for secure sharing of PHR based on Lewko and Waters's scheme, in which an anonymous secret key issuing protocol is used to hide GID. This protocol can make the authorities generate the correct decryption key for users without knowing their GID. In addition, the one-way anonymous key agreement is used to hide the attributes in the access policy. The presented scheme keeps the security of the Lewko and Waters's scheme

and removes the random oracle, but, computational cost is high.

2) HIDDEN ACCESS STRUCTURE/POLICY ABE

In general CP-ABE schemes, the ciphertext is associated with access policy generally carrying sensitive information of data owner (DO), and the public access policy of ciphertext is likely to lead to disclosure of sensitive information of DO. Therefore, in order to ensure the user's privacy. Phuong *et al.* [29] proposed an attribute-based encryption scheme with hidden access policy. This construction uses an access policy with only AND gates. In 2015, Xu *et al.* [30] extended the ABE scheme proposed by Bethencourt *et al.* [2] with the hidden access policy feature for cloud applications. However, this ABE scheme relies on the use of a CA to manage all the attributes and private keys in the system. Later, Zhong *et al.* [31] proposed the first policy hidden attribute-based encryption scheme using multi-authority architecture. However, this scheme introduces an expensive computation cost at the client side to execute the decryption process. Belguith *et al.* [32] presented a novel Policy-Hidden Outsourced Attribute-Based Encryption (PHOABE) scheme, in which the user outsources the expensive computations. Unfortunately, it requires a special CA. In 2013, Qian *et al.* [33] proposed a privacy-preserving decentralized CP-ABE with Fully Hidden Access Structure. Although the authorities can get nothing about user GID when generating and issuing user private keys and access structures are hidden to receivers, which uses an access structure with limited expressiveness.

3) HIERARCHICAL ABE

In some applications, a DO wants to share multiple files, multiple access policies are required, which can be complex and take up a lot of storage space. That is to say, as the number of files increases, ciphertext storage and computation overload will increase. To solve this problem, the Gentry *et al.* [34] firstly proposed the concept of hierarchical encryption, which mainly used identity-based encryption (IBE). Successively, many hierarchical CP-ABE schemes have been proposed. Wang *et al.* [35] proposed a hierarchical CP-ABE scheme by combining the hierarchical IBE [34] and the CP-ABE. Wan *et al.* [36] proposed a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. Although the scheme used a multi-authority to manage the distribution key, a special CA must exist. Later, Shen *et al.* [37] proposed an authentication scheme in cloud, in which a hierarchical attribute authorization structure was used. However, a trusted root authority was needed. Wang *et al.* [21] proposed an ABE scheme based on file hierarchical in cloud computing. Unfortunately, this scheme only has one CA manages the key and there is no policy hiding, which is easy to cause dishonest attribute authority key escrow attack and the disclosure of user privacy.

II. PRELIMINARIES

A. BILINEAR MAPS AND DBDH ASSUMPTION

We describe some preliminaries and other useful concepts that are used in our approach in this section.

Bilinear Maps: Let G_0 and G_T be two groups of prime order p. The generator of G_0 is g. A bilinear mapping $e : G_0 \times G_0 \rightarrow G_T$ satisfies the following properties:

(1) Bilinearity: For any $g_1, g_2 \in G_0$ and $a, b \in Z_p$, it has $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.

(2) Non-degeneracy: There exists $g_1, g_2 \in G_0$ such that $e(g_1, g_2) \neq 1$.

(3) Computability: For all $g_1, g_2 \in G_0$ and $a, b \in Z_p$, there is an efficient computation $e(g_1, g_2)$.

DBDH Assumption: Let G_0 be a group with prime order p, g be a generator in G_0. We say that DBDH assumption holds if no probabilistic polynomial time (PPT) adversary can distinguish the tuples $(g, A = g^a, B = g^b, C = g^c, e(g, g)^{abc})$ and $(g, A = g^a, B = g^b, C = g^c, e(g, g)^d)$, where $a, b, c, d \in Z_p$. The advantage of algorithm $\Re$ is

$$Adv_B^{DBDH} = |\Pr[\Re(A, B, C, e(g, g)^{abc}) = 0] - \Pr[\Re(A, B, C, e(g, g)^d) = 0]|$$

B. ACCESS STRUCTURE

The description is similar as the literature [2]. Let $\{p_1, p_2, \cdots, p_n\}$ be a set of parties. A collection is monotone if $\forall B, C$: if $B \in A$ and $B \subseteq C$ then Γ. An access structure (respectively, monotone access structure) is a collection A of non-empty subsets of l, i.e. $A \subseteq 2^{\{p_1, p_2, \cdots p_n\}} \backslash \{\emptyset\}$. The sets in A are called the authorized sets, otherwise, the sets are called the unauthorized sets Γ.

Generally, the data users are described by attributes. The authorized sets are included in A.

C. HIERARCHICAL ACCESS TREE

Hierarchical tree is an integrated access structure by multiple access structure, which can provide multiple hierarchical files sharing.

Let Γ be a hierarchical tree representing an access structure which is divided into l access levels. Nodes of the tree are denoted as (x, y). The symbol x represents the node's row in Γ (from top to bottom), and y represents the node's column in Γ(from left to right). In Figure. 2, the nodes can be denoted as: $R = (1, 1)$, $A = (2, 1)$, $B = (2, 2)$, $C = (3, 1)$, $D = (3, 2)$, $E = (4, 1)$, $F = (4, 2)$, $G = (5, 1)$, $H = (5, 2)$, $I = (5, 3)$, To facilitate description of the access tree, several functions and terms are defined as follows.

(1) (x, y): It denotes a node of tree Γ. If (x, y) is a leaf node, it denotes an attribute. If (x, y) is a non-leaf node, it denotes a threshold gate, such as "AND", "OR", "n-of-m (n < m)". For example, the nodes E and G denote a threshold gate and an attribute in Figure. 2.

(2) $num_{(x,y)}$: It denotes the number of (x, y)'s children in Γ. For example, $num_R = 2$ in Figure. 2.

(3) $k_{(x,y)}$: It denotes threshold value of node (x, y), where $1 \leq k_{(x,y)} \leq num_{(x,y)}$. When $k_{(x,y)} = 1$ and (x, y) is a non-leaf

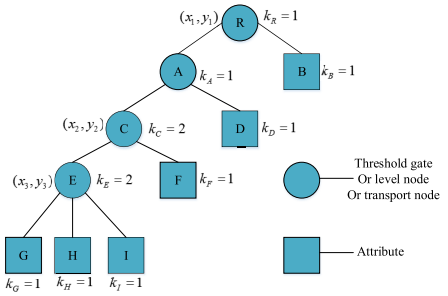

FIGURE 2. An example of three-level access tree.

node, (x, y) is an OR gate. When $k_{(x,y)} = num_{(x,y)}$ and (x, y) is a non-leaf node, it is an AND gate. In particular, if (x, y) is a leaf node $k_{(x,y)} = 1$. For example, $k_C = 2$ denotes an AND gate in Figure. 2.

(4) $(x_i, y_i)(i \in [1, l])$: It denotes level node of Γ. In this work, access tree Γ is divided into l access levels. And the hierarchy of the nodes is sorted in descending order. That is, (x_1, y_1) is the highest hierarchy, and (x_l, y_l) is the lowest hierarchy. For example, $(x_3, y_3) = E$ is the Third hierarchy in Figure. 2.

(5) $parent(x, y)$: It represents the parent of the node (x, y) in Γ. For example $parent(2, 1) = parent(A) = R$, $parent(3, 1) = parent(C) = A$ in Figure. 2.

(6) transport node: The node (x, y) is a transport node if one of the children of (x, y) contains at least one threshold gate. For example, R, A and C are transport nodes in Figure. 2.

(7) $TN - CT(x, y)$: It represents a threshold gate set of transport node (x, y)'s children in Γ. It is marked as $TN - CT(x, y) = \{child_1, child_2, \cdots\}$. For example $TN - CT(R) = \{A\}, TN - CT(A) = \{C\}, TN - CT(C) = \{E\}$ in Figure. 2.

(8) $att(x, y)$: It denotes an attribute associated with the leaf node (x, y) in Γ.

(9) $index(x, y)$: It returns an unique value associated with the node (x, y), where the value is assigned to (x, y) for a given key in an arbitrary manner.

(10) Γ_R: It denotes a tree diagram, where root node of the tree is R.

(11) $\Gamma_{(x,y)}$:It denotes the sub-tree of Γ rooted at the node (x, y). If an attribute set S satisfies $\Gamma_{(x,y)}$, we denote it as $\Gamma_{(x,y)}(S) = 1$. $\Gamma_{(x,y)}(S)$ is recursively computed as follows. If (x, y) is a non-leaf node, $\Gamma_{(x,y)}(S)$ returns 1if and only if at least $k_{(x,y)}$ children return 1. If (x, y) is a leaf node, then $\Gamma_{(x,y)}(S)$ returns 1 if and only if $att(x, y) \in S$.

D. ONE-WAY ANONYMOUS KEY AGREEMENT

In 2007, Kate *et al.* proposed a one-way anonymous key agreement scheme (2007) [38], which can guarantee anonymity for each participant. Suppose Alice (ID_A) and Bob (ID_B) are two participants, the master key of key generation center (KGC) is s. Alice wants to keep anonymity with Bob, the progress of key agreement protocol is as follows:

(1) Alice calculates $Q_B = H(ID_B)$. She randomly selects a number $r_A \in Z_p^*$ to generate the pseudonym $P_A = Q_A^{r_A}$ and calculates the session key $K_{A,B} = e(d_A, Q_B)^{r_A} = e(Q_A, Q_B)^{sr_A}$. Finally, she sends her pseudonym P_A to Bob.

(2) Bob calculates the session key $K_{A,B} = e(p_A, d_B) = e(Q_A, Q_B)^{sr_A}$ using his secret key d_B, where $d_i = H(ID_i)^s \in G_1$ is user's private key for $i \in \{A, B\}$, and $H : \{0, 1\}^* \rightarrow G_0$ is a strong collision-resistant hash function.

III. PROBLEM FORMULATIONS

A. SYSTEM MODEL

In our system model, data owners are patients, data users (such as doctors, nurses, other patients, family and medical researches) are different and have different access permissions (such as access to general medical records, access to all medical records) for with different attributes (such as attending doctor, cardiovascular department, chief physician, supervisor and system administrator).

As shown in Figure. 3, there are four main entities in the system: the multiple attribute authority (A_j), cloud server (CSP), the data Owner (DO) and data User (DU), among which:

(1) *Multiple attribute authorities* (A_j). Each A_j is fully trusted, manages different set of attributes, and generates anonymous private key for each user through interaction with the user. They work separately and do not interact with each other. In this scheme, they mainly perform two algorithms *Authoritysetup* and *KeyGen*.

(2) *Cloud service provider* (CSP). It is a semi-trusted entity associated with modern medical system. It can honestly perform the assigned tasks and return correct results. However, it would like to find out as much sensitive (x, y) contents as possible. In the proposed system, it provides encrypted EHRs storage and transmission services.

(3) *The data Owner* (DO). He defines the access structure, hides access structure, define the authentication tag, collects physiological data (such as blood pressure, pulse and heart rate), and uploads the encrypted EHRs and authentication tag to the CSP.

(4) *Data users* (DU). As a cloud user, the DU obtains the corresponding symmetric key AK_k if his attributes set meets the access structure, then he obtains the corresponding plaintext EHR.

Different authority A_j manages different attribute set of DU and distributes the corresponding private key SK_j. Suppose that a DO wants to outsource two files $M = \{m_1, m_2\}$ in cloud, where m_1 is the highest level of access on the tree and m_2 is the lowest level, he encrypts $M = \{m_1, m_2\}$ based on hierarchical tree. If the DU's attributes set T^j satisfying the access structure Γ, he can obtain the corresponding symmetric key AK_k. In the multi-level access structure, DUs with different attribute sets will get different symmetric keys. As shown in Figure 3, DU1 satisfies the whole access level Γ, so he obtains AK_1 and AK_2; DU2 only meets part of the access level, then he obtains AK_2; DU3 does not satisfy the access structure and cannot get any symmetric key, so the decryption fails.

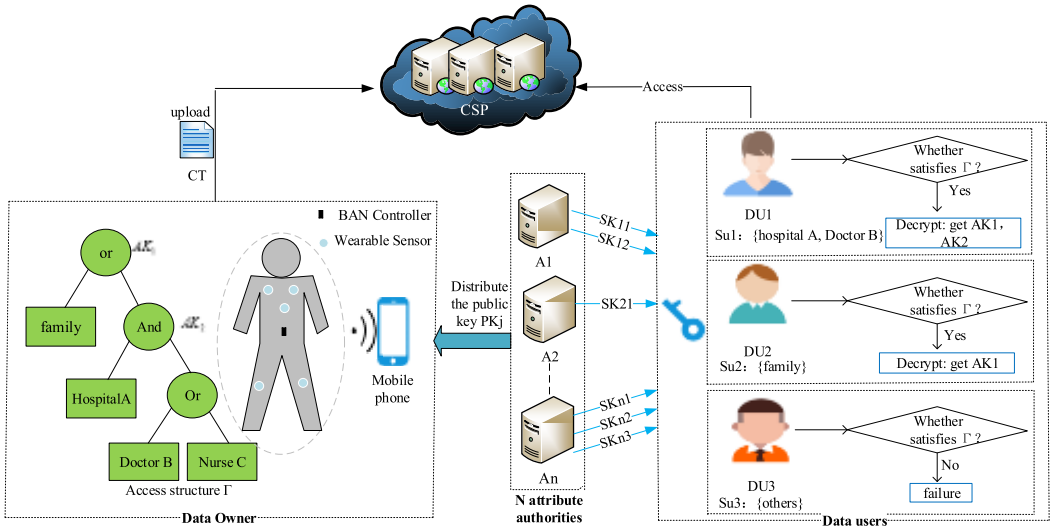

FIGURE 3. Our EHR sharing system model.

TABLE 1. Notations.

Notations	Description
GID	Global identifier of DU
A_j	Attribute authority j
$\tilde{A}_j$	an attribute set of A_j monitors
MSK_j	Master secret key
SK_j	Private keys of DU from A_j
PK_j	Public keys
T	The set of attributes associated with GID
T'	The set of attributes associated with the DO
AK_k	The corresponding symmetric key
S_u^j	Hidden set of attributes associated with the DU
T^j	The set of attribute associated with SK_j
M	The set of plaintext
l	The number of files
m_k	The corresponding plaintext
Γ	The hierarchical access tree
V	The set of verification tag
V_k	Each item of verification tag V
γ_i	Replaced attributes i in the access tree Γ
S_o	Hidden attributes set associated with the DO
I_c	An index set of authorities A_j

Notations used in the rest of the paper are summarized in Table 1.

B. DECENTRALIZED HIERARCHICAL ABE SCHEME

A decentralized hierarchical ABE scheme consists of the following five algorithms:

Global setup$(\lambda) \rightarrow GP$. This algorithm takes a security parameter λ as input and returns the system parameters GP.

Authority setup$(GP) \rightarrow (MSK_j, PK_j)$. Each authority A_j runs this algorithm to generate his master secret key MSK_j and public key PK_j for $j = 1, 2, \cdots, N$, where N is the number of totally authorities in the system.

KeyGen$(PK_j, GID, GP, T^j) \rightarrow SK_j$. Each authority A_j takes as input his public keys PK_j, a global identifier GID of a user DU, system parameters GP and a set of attributes T^j, outputs a private key SK_j for user.

Encrypt$(GP, M, PK_j) \rightarrow CT$. This algorithm takes as input system parameters GP, l messages $M = \{m_1, m_2, \cdots, m_l\}$,hierarchical access structure, public keys $PK_j (j \in I_c)$, I_c represents the set of attribute authorities A_j involved. It outputs the ciphertext CT.

Decrypt$(GP, CT, SK_j, PK_j) \rightarrow m_k$. This algorithm takes as input system parameters GP, the ciphertext CT, public key PK_j and private key SK_j associated with attribute set T^j, if the user's attributes set meets partial access level, the corresponding plaintext m_k can be obtained. If the user satisfies the entire access level Γ, the desired plaintext M can be obtained completely, and then decrypts $M = \{m_1, m_2, \cdots, m_l\}$ successively.

C. SECURITY MODEL

The security model for N-authority decentralized hierarchical ABE scheme is defined as follows:

Init: Adversary $\mathcal{A}$ submits the challenge access structures Γ_0^*, Γ_1^* and a list of corrupted authorities C_A to algorithm $\Re$, where $|C_A| < N$.

Setup: The challenger $\mathcal{C}$ runs the algorithm *Globalsetup* and outputs the system parameters GP to adversary $\mathcal{A}$.

Authority Setup: For the corrupted authorities, the challenger sends his public and secret keys (PK_j, SK_j) to the adversary $\mathcal{A}$. For the honest authorities, the challenger $\mathcal{C}$ sends his public keys PK_j.

Phase 1: The adversary $\mathcal{A}$ sends an attribute list T to the challenger $\mathcal{C}$ for secret keys queries, where $\mathrm{T} \nvDash \Gamma_0^*$ and $\mathrm{T} \nvDash \Gamma_1^*$. This process can be repeated q times.

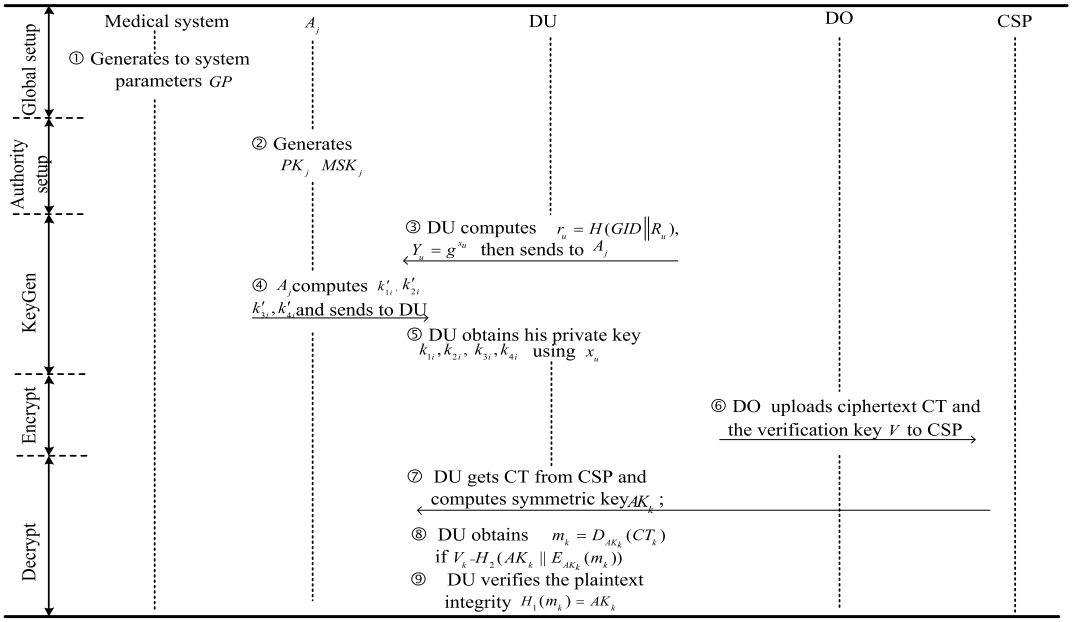

FIGURE 4. Overview of the proposed scheme.

Challenge: The adversary $\mathcal{A}$ submits two equal length messages m_0 and m_1. The challenger $\mathcal{C}$ chooses a random bit $\mu \in \{0, 1\}$ and runs the algorithm Encryption to obtain ciphertext CT^*_μ. The challenger $\mathcal{C}$ returns the ciphertext CT^*_μ to the adversary $\mathcal{A}$.

Phase 2: Phase 1 is repeated adaptively.

Guess: Finally adversary $\mathcal{A}$ outputs his guess $\hat{\mu} \in \{0, 1\}$. If $\hat{\mu} = \mu$, $\mathcal{A}$ wins the security game. In this game, $\mathcal{A}$ can win the game which is defined as $Pr[\mu = \hat{\mu}] - 1/2|$.

Definition 1: A N-authority this scheme is secure in the above security model if no probabilistic polynomial-time adversary $\mathcal{A}$ making q secret key queries has advantage at least $Adv^{N-CP-ABE}(1^\lambda) = |Pr[\mu = \hat{\mu}] - 1/2| > \varepsilon(\lambda)$ in the above security model.

IV. ANONYMOUS HEALTH RECORD SHARING SCHEME BASED ON DECENTRALIZED HIERARCHICAL ABE

In this section, we first give an overview of our scheme shown in Figure 4, then present the concrete construction in Section 4.1.

A. CONCRETE CONSTRUCTION

In this part, we will present our detailed construction of our scheme.

(1) *Global setup*$(\lambda) \rightarrow (GP)$. Given the security parameter λ, the algorithm returns a system parameter $(e, g, p, G_0, G_T, H, H_0, H_1, H_2)$, where, G_0, G_T are the multiplication cycle groups with prime order p, a bilinear mapping $e : G_0 \times G_0 \rightarrow G_T$, and g is the generator of group G_0. Suppose there are N authorities in the system, namely $A_1, A_2, \cdots, A_N$, which A_j monitors a set of attributes $i \in \tilde{A}_j$, and $\tilde{A}_j \cap \tilde{A}_{j'} = \phi$, if $j \neq j'$. For any $k \in Z_p$ and an attribute set $\tilde{A}_j = \{a_{j,1}, a_{j,2}, \cdots, a_{j,n_j}\}$, the Lagrange coefficient $\Delta_{k,\tilde{A}_j} = \Pi_{l \in \tilde{A}_j, l \neq k}(x - l)/(k - l)$. The four non-collision hash functions are $H : \{0, 1\}^* \rightarrow G_0, H_0 : \{0, 1\}^* \rightarrow G_T, H_1 : \{m_k\} \rightarrow \{0, 1\}^{n_{H_1}}, H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^{n_{H_2}}$, respectively.

(2) *Authority setup*$(GP) \rightarrow (PK_j, MSK_j)$. In this system, for any A_j running the algorithm to generate the corresponding PK_j and MSK_j, where $j = 1, 2, \cdots N$. A_j randomly selects $\alpha_j \in Z_p, \beta_{ji} \in Z_p (i \in [1, n_j])$, then outputs the PK_j and MSK_j as following:

$$PK_j = \{e(g, g)^{\alpha_j}, \{h_{ji} = g^{\beta_{ji}}\}_{\forall i \in \tilde{A}_j}\}, MSK_j = \{g^{\alpha_j}, \{\beta_{ji}\}_{\forall i \in \tilde{A}_j}\}$$

(3) *KeyGen*$(GID, T^j, PK_j, GP) \rightarrow SK_j$. In this subsection, to generate the secret key for user, the attribute authority A_j executes an interactive process with the user.

- The user DU randomly chooses $R_u \in Z_P, x_u \in Z_P$ to compute $r_u = H(GID \| R_u), Y_u = g^{x_u}$, where GID is the global identity of the user. And then sends (r_u, Y_u) to the A_j.
- Next the A_j selects a random number $r_j \in Z_p, r_i \in Z_p^*$ to calculate $k'_{1i} = Y_u^{\alpha_j} h_{ji}^{r_i}, k'_{2i} = Y_u^{r_j} H(i)^{r_i/(r_u+\beta_{ji})}$, $k'_{3i} = h_{ji}^{r_i/(r_u+\beta_{ji})}$ and $k'_{4i} = H(i)^{\beta_{ji}}$, then returns $(k'_{1i}, k'_{2i}, k'_{3i}, k'_{4i})$ to DU. Here $T^j = \tilde{A}_j \cap T$, where T represents the set of attributes associated with the user's GID, and for any attribute $i \in T^j$
- DU obtains his private key using x_u as follows:

$$k_{1i} = (k'_{1i})^{1/x_u} = g^{\alpha_j} h_{ji}^{r_i/x_u}, k_{2i} = (k'_{2i})^{1/x_u} = g^{r_j} H(i)^{r_i/x_u(r_u+\beta_{ji})},$$
$$k_{3i} = (k'_{3i})^{1/x_u} = h_{ji}^{r_i/x_u(r_u+\beta_{ji})}, k_{4i} = k'_{4i} = H(i)^{\beta_{ji}},$$

where, the private key k_{4i} is used for attribute hiding. Finally, the private is:

$$SK_j = \{k_{1i} = g^{\alpha_j} h_{ji}^{r_i/x_u}, k_{2i} = g^{r_j} H(i)^{r_i/x_u(r_u+\beta_{ji})}, k_{3i} = h_{ji}^{r_i/x_u(r_u+\beta_{ji})}, k_{4i} = H(i)^{\beta_{ji}}\}_{\forall i \in T^j}$$

(4) $Encrypt(PK_j, GP, AK, \Gamma) \rightarrow CT$. Assume that a DO want to share l files, i.e., $M = \{m_1, m_2, \cdots, m_l\}$, with l access levels. Then, the symmetric key is $AK_k = H_1(m_k)$ are defined and get the symmetric key $AK = \{AK_1, AK_2, \cdots, AK_l\}$, where $AK_k = H_1(m_k)$, $k = 1, 2, \cdots, l$. The DO encrypts the files using symmetric encryption algorithm *AES* as following: $CT_{AK} = \{E_{AK_1}(m_1), E_{AK_2}(m_2), \cdots, E_{AK_l}(m_l)\}$. Let $V = \{V_1, V_2, \cdots, V_l\}$ be the verification tag, where $V_k = H_2\left(AK_k \| E_{AK_k}(m_k)\right), k = 1, 2, \cdots, l$. The DO sets a hierarchical access tree Γ. For each attribute $i \in \Gamma$, the DO computes $\gamma_i = e((h_{ji})^{\tau}, H(i))$, and get $S_o = \{\gamma_i : i \in T'\}$, where $\tau \in Z_p$ is a random exponent, and T' represents a hid set of attributes associated with the DO. Namely, the DO uses γ_i to replace the attribute i in the hierarchical access tree Γ. Finally, the algorithm outputs ciphertext CT and the verification tag V. The details are as follows:

- DO sets level nodes $(x_k, y_k)(k = 1, 2, \cdots, l)$ in Γ, and selects l random numbers $s_1, s_2, \cdots, s_l \in Z_p$. Then, he computes $\tilde{C}_k$ and C'_k for all $k = 1, 2, \cdots, l$ as follows:

$$\tilde{C}_k = \prod_{j \in I_c} AK_k e(g, g)^{\alpha_j s_k}, \quad C'_k = g^{s_k},$$

where I_c is an index set of authorities A_j, that is, a level nodes relates of has multiple attribute authorities simultaneously.

- Polynomial structure rule: a polynomial $q_{(x,y)}$ needs to be selected for each node (x, y) (including the leaf nodes) in Γ. From the root node R, the polynomial $q_{(x,y)}$ is randomly selected from top to bottom manner. For each node (x, y) in Γ, degree of the polynomial $d_{(x,y)} = k_{(x,y)} - 1$, where $k_{(x,y)}$ is the threshold value.
- Beginning from the root node R, data owner sets $q_R(0) = q_{(x_1,y_1)}(0) = s_1$ and chooses d_R other points of the polynomial q_R to define it completely, where the points are made of two types of nodes. The ones are level nodes which are children of R. The others are remaining nodes randomly selected. For each non-root node (x, y), he sets $q_{(x,y)}(0) = q_{(x_k,y_k)}(0) = s_k$ if the (x, y) is a level node. Otherwise, $q_{(x,y)}(0) = q_{parent(x,y)}(index(x, y))$. The other $d_{(x,y)}$ points of $q_{(x,y)}$ are a made of the level nodes of the children of (x, y) and the remaining nodes randomly selected.
- Ciphertext at leaf nodes: Let Y be the set of leaf nodes in Γ. For each node $(x, y) \in Y$, ciphertext is computed as follows:

$$C_{(x,y)} = h_{ji}^{q_{(x,y)}(0)}, C'_{(x,y)} = H(att(x, y))^{q_{(x,y)}(0)}, C''_{(x,y)} = g^{\tau}.$$

- Transport nodes ciphertext: In Γ, let X be the set of transport nodes, and $TN - CT(x, y)$ be the threshold gate set of transport node (x, y)'s children, where $TN - CT(x, y) = \{child_1, child_2, \cdots, child_{k'}, \cdots\}$. Then, The DO computes $\hat{C}_{(x,y),k'}(k' = 1, 2, \cdots)$ for each node (x, y) in the set of X as follows:

$$\hat{C}_{(x,y),k'} = \{e(g, g)^{\sum_{j \in I_c} \alpha_j (q_{(x,y)}(0) + q_{(x,y),k'}(0))} \cdot H_0(e(g, g)^{\sum_{j \in I_c} \alpha_j q_{(x,y)}(0)})\}.$$

To sum up, the data owner finally output the complete ciphertext CT is as follows:

$$CT = \{\Gamma, C_{AK}, \{\tilde{C}_k, C'_k\}_{\forall k \in [1,l]}, \{C_{(x,y)}, C'_{(x,y)}, C''_{(x,y)}\}_{\forall (x,y) \in Y}, \{\hat{C}_{(x,y),k'}\}_{\forall k' \in 1,2,\ldots}\}.$$

(5) $Decrypt(PK_j, CT, GP, SK_j) \rightarrow AK_k$. For any $i \in T^j$, DU computes $\gamma'_i = e(C''_{(x,y)}, k_{4i}) = e(g^{\tau}, H(i)^{\beta_{ji}})$ and let $S_u^j = \{\gamma'_i : i \in T^j\}$. If $S_u^j \subseteq S_o$, the user's attribute sets satisfies hierarchical access tree Γ, so, the corresponding symmetric key AK_k can be obtained. Then, similar to CP-ABE [2], a recursive operation $DecryptNode(CT, SK, (x, y))$ should be first defined.

- If (x, y) is a leaf node, we let $i = att(x, y)$ and define $DecryptNode(CT, SK_j, (x, y))$ as below. If $i \notin T^j$, $DecryptNode(CT, SK_j, (x, y)) = null$. Otherwise, the operation $DecryptNode(CT, SK_j, (x, y))$ is obtained as follows:

$$\begin{aligned} &DecryptNode(CT, SK_j, (x, y)) \\ &= \frac{e(k_{2i}, C_{(x,y)})}{e(k_{3i}, C'_{(x,y)})} \\ &= \frac{e(g^{r_j} H(i)^{r_i/x_u(r_u+\beta_{ji})}, h_{ji}^{q_{(x,y)}(0)})}{e(h_{ji}^{r_i/x_u(r_u+\beta_{ji})}, H(att(x, y))^{q_{(x,y)}(0)})} \\ &= e(g, g)^{r_j \beta_{ji} q_{(x,y)}(0)} \end{aligned}$$

- If (x, y) is a non-leaf node, $DecryptNode(CT, SK, (x, y))$ is defined as below. For all nodes z that are children of (x, y), the DU runs $DecryptNode(CT, SK_j, z)$ and stores the output as F_z. Let $s_{(x,y)}$ be an arbitrary $k_{(x,y)}$- sized child nodes set z, and then $F_z \neq null$. If the set does not exist, $F_z = null$. Otherwise, $F_{(x,y)}$ is computed as follows, where $s'_{(x,y)} = \{index(z) : z \in s_{(x,y)}\}$, $i = index(z)$.

$$\begin{aligned} F_{(x,y)} &= \prod_{z \in s_{(x,y)}} F_z^{\Delta_{k,s'_{(x,y)}(0)}} \\ &= \prod_{z \in s_{(x,y)}} (e(g, g)^{\sum_{j \in I_c} r_j \beta_{ji} q_z(0)})^{\Delta_{k,s'_{(x,y)}(0)}} \\ &= e(g, g)^{\sum_{j \in I_c} r_j \beta_{ji} q_{(x,y)}(0)} \end{aligned}$$

Next, the procedures of decryption are divided into four steps:

Step 1, if the attribute set T satisfies part or the whole Γ, that is, T satisfies part or the whole level nodes

(x_k, y_k), $e(g,g)^{\sum_{j\in I_c} r_j\beta_{ji}s_k}$, $(k \in [1,l])$ can be obtained by the recursive operation of the formula as follow:

$$\begin{aligned} A_k &= DecryptNode(CT, SK_j, (x_k, y_k)) \\ &= e(g,g)^{\sum_{j\in I_c} r_j\beta_{ji}q_{(x_k,y_k)}(0)} \\ &= e(g,g)^{\sum_{j\in I_c} r_j\beta_{ji}s_k} \end{aligned}$$

Step 2, $e(g,g)^{\sum_{j\in I_c}\alpha_j s_k}$ can be computed by the formula as follow:

$$F_k = \frac{e(C'_k, \prod_{j\in I_c} k_{1i})}{(A_k)^{1/x_u}} = \frac{e(g^{s_k}, \prod_{j\in I_c} g^{\alpha_j} h_{ji}^{r_j/x_u})}{e(g,g)^{\sum_{j\in I_c} r_j\beta_{ji}s_k \Big/ x_u}} = e(g,g)^{\sum_{j\in I_c}\alpha_j s_k}$$

Based on the hierarchical nodes, if T includes the lower authorization nodes, we can recursively calculate all of the authorization's level nodes with the values of transport nodes $\hat{C}_{(x,y),k'}(k' = 1, 2, \cdots)$ by using the formula as follow. Therefore $F_{k+1,k'}, \cdots, F_{l,k'}$ are obtained sequentially. That is, the values $e(g,g)^{\sum_{j\in I_c}\alpha_j s_k}, \cdots, e(g,g)^{\sum_{j\in I_c}\alpha_j s_l}$ are got.

$$\begin{aligned} F_{k+1,k'} &= \frac{\hat{C}_{(x_k,y_k),k'}}{F_k \cdot H_0(F_k)} \\ &= \frac{e(g,g)^{\sum_{j\in I_c}\alpha_j(s_k + q_{child_{k'}}(0))} \cdot H_0(e(g,g)^{\sum_{j\in I_c}\alpha_j s_k})}{e(g,g)^{\sum_{j\in I_c}\alpha_j s_k} \cdot H_0(e(g,g)^{\sum_{j\in I_c}\alpha_j s_k})} \\ &= e(g,g)^{\sum_{j\in I_c}\alpha_j q_{child_{k'}}(0)} \end{aligned}$$

Step 3, the corresponding symmetric key $AK_k(k \in [1,l])$ are decrypted by executing the formula as follow repeatedly.

$$\frac{\tilde{C}_k}{F_k} = \frac{\prod_{j\in I_c} AK_k e(g,g)^{\alpha_j s_k}}{e(g,g)^{\sum_{j\in I_c}\alpha_j s_k}} = AK_k(*)$$

Step 4, the user first uses the verification key V_k to verify the integrity of the ciphertext, for any, if the algorithm verify $V_k \neq H_2\left(AK_k \left\| E_{AK_k}(m_k)\right.\right)$, the algorithm will be terminated immediately; Otherwise, use equation $(*)$ above to get the corresponding symmetric key AK_k, then use the symmetric key AK_k to decrypt $D_{AK_k}(CT_k)$, so as to get plaintext m_k. Then the user uses the hash function H_1 to authenticate plaintext m_k, if $H_1(m_k) = AK_k$, user successfully obtain correct plaintext information m_k; Otherwise, the plaintext information been tampered with and the acquisition fails.

V. SECURITY PROOF

Theorem 1: Suppose DBDH assumption holds. Then no polynomial adversary can selectively break the proposed N-authorities this scheme.

Proof: As is show in Figure. 5, Challenger $\mathcal{C}$ and adversary $\mathcal{A}$ play the following 9 times interactive game, mainly including Initialization, Authorities setup, QueryPhase 1, Challenge, QueryPhase 2, Guess and so on.

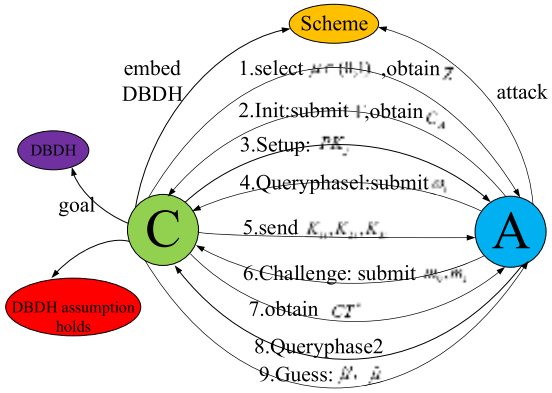

FIGURE 5. Overview of the theorem proving.

In conclusion, the DBDH hypothesis is valid. That is, if DBDH assumption holds, then no polynomial adversary can selectively break the proposed N-authorities this scheme.

The above detailed proof procedure of theorem 1 is shown in appendix

Theorem 2: The proposed scheme can resist the collusion attack of users.

Proof: When DU requests private key from attribute authority A_j, he firstly presents $r_u = H(GID \| R_u)$, where GID is a unique global identity and $R_u \in Z_P$ is a random number of DU. Secondly, A_j chooses $r_j \in Z_p$, $r_i \in Z_p^*$ randomly to computer$(k'_{1i}, k'_{2i}, k'_{3i}, k'_{4i})$ for DU. At last, the user's private key is $\{k_{1i} = g^{\alpha_j} h_{ji}^{r_i/x_u}, k_{2i} = g^{r_j} H(i)^{r_i/x_u(r_u+\beta_{ji})}$, $k_{3i} = h_{ji}^{r_i/x_u(r_u+\beta_{ji})}, k_{4i} = H(i)^{\beta_{ji}}\}_{\forall i\in T^j}$. If two DUs with different identity GID and GID' want to make a collusion attack by combining their keys, then it will appear some terms in the forms of $k_{1i} = g^{\alpha_j} h_{ji}^{r_i/x_u}, k_{2i} = g^{r_j} H(i)^{r_i/x_u(H(GID\|R_u)+\beta_{ji})}, k_{3i} = h_{ji}^{r_i/x_u(H(GID\|R_u)+\beta_{ji})}$ and other terms of $k_{1i} = g^{\alpha_j} h_{ji}^{r'_i/x_{u'}}, k_{2i} = g^{r'_j} H(i)^{r'_i/x_{u'}(H(GID'\|R_{u'})+\beta_{ji})}$, $k_{3i} = h_{ji}^{r'_i/x_{u'}(H(GID'\|R_{u'})+\beta_{ji})}$ during the decryption process. Therefore, our scheme can resist collusion attack.

Theorem 3: The proposed scheme can resist the multiple authorities' collusion to obtain user's private key by tracing the global identity GID.

Proof: To protect the user's private key, anonymous key generation is introduced in our scheme during the *KeyGen* phase. Firstly, the A_j gets only $r_u = H(GID \| R_u)$, while not the global identity GID. Secondly, once data user DU receiving the private key $(k'_{1i}, k'_{2i}, k'_{3i}, k'_{4i})$ form A_j, he obtains the real private key $(k_{1i} = (k'_{1i})^{1/x_u}, k_{2i} = (k'_{2i})^{1/x_u}, k_{3i} = (k'_{3i})^{1/x_u})$ by using his random number x_u. So each A_j neither knows the DU's real identity nor the corresponding private key.

Therefore, our scheme can resist the multiple authorities' collusion to obtain user's private key by tracing the GID.

VI. PERFORMANCE ANALYSIS

In this section, we will conduct a comprehensive performance evaluation between our scheme and other related

TABLE 2. Notations.

Notations	Description
p_e	Pairing operation
e_T	Exponentiation in G_T
e_0	Exponentiation in G_0
S_i	The least interior nodes satisfying Γ
A_u	The attributes set of user u
A_{C_i}	The attributes related with ciphertext
A_T	The set of transport nodes
$\|*\|$	The number of elements in $*$.
L_*	The bit-length of element in $*$
l	The number of files

schemes [2], [21], [28], [33], [37] from the aspects of functions, computational and storage cost. Table 2 shows the notations used in performance analysis.

A. THEORETICAL ANALYSIS

1) FUNCTIONAL COMPARISON

Table 3 shows the comparison of functions between our scheme with schemes [2], [21], [28], [33], [37]. Amongst them, schemes [2], [28], [33] are all CP-ABE type. The access structure of these type of schemes will increase linearly with the number of files. Thus, ciphertext also increases linearly, thereby occupying a large amount of storage space in the cloud. Compared with the hierarchical attribute based signature (HABS) scheme [37], our scheme is introduced decentralized multi-authority that solve the bottleneck and monopoly of centralized authority, and the global identifier GID is introduced to resist collusion attacks. Additionally, our scheme makes a double verification of ciphertext which ensures information correctness and integrity. But scheme [37] has no such features. Moreover, our scheme and scheme [21] belong to the FH-CP-ABE type, in which only one access policy is needed to encrypt multiple files so that it greatly saves the cloud storage space. In addition, compared with other access structures, hierarchical access tree structure has the characteristics of more fine-grained access control. However, in scheme [21], this structure cannot launch the correctness verification of the message, which may easily lead to the possibility of message tampering. Secondly, the data users do not hide their identities when encrypting messages; as such, dishonesty in authority tracking attack may exist. Finally, the scheme is only applicable to the distribution of keys by AA management, thereby making it vulnerable to dishonest AA key escrow attacks. Thus, our scheme is functional.

2) COMPUTATIONAL COST

To facilitate analysis, we assume that the set $TN - CT(x, y)$ of each transport node (x, y) contains k' nodes, that is, $TN - CT(x, y) = (child_1, child_2, \cdots, child_{k'})$. The number of the nodes in the set $TN - CT(x, y)$ is considered in the following analysis because they are closely related to the computation and storage cost of transport node, which is associated with encryption, decryption and ciphertext CT. Suppose that there are l hierarchical files, that is $M = \{m_1, m_2, \cdots, m_l\}$, and their access order is decreased. Thus, the attributes can be denoted as $\{A_{C_1}, A_{C_2}, \cdots, A_{C_l}\}$, where $A_{C_1} \supseteq A_{C_2} \supseteq \cdots \supseteq A_{C_l}$.

According to the data in Table 4, the cost of encryption and decryption of scheme [2], [28], [33] are too expensive because of the increase in the number of shared files and attribute sets related with ciphertexts. Compared with scheme [2], [28], [33], our scheme saves time in encryption and decryption significantly.

3) STORAGE COST

The comparison of storage cost is shown in Table 5. The size of the PK, MSK and SK of our scheme, scheme [2], scheme [28] and scheme [33] are shown. In our scheme the keys occupies less storage space. In addition, with the increase in the number of files and attributes, ciphertext storage overhead in our scheme is more acceptable compared with schemes [2], [28], [33] in terms of ciphertext storage.

B. EXPERIMENT SIMULATION

We implement our scheme with the related works on a laptop with 64-bit Windows 8 operating system with 2.39 GHZ Intel (R) core (TM) i5-4210u CPU with 8GB RAM. All algorithms are done in the C language and our codes use the pairing-base cryptography (PBC) library [39]. Concretely, we select the type an elliptic curve parameter with the 160-bit order. The following is simulation results of encryption and decryption time and ciphertext storage size with increase of the number of attributes and file in our scheme and schemes [2], [28], [33] respectively. Figure 6 shows that with the increase in the number of attributes, the time efficiency of our scheme is obviously higher during the encryption and decryption phases for $l = 4$. Figure 7 shows that our scheme can reflect the superiority of its time efficiency better with the increase in the number of files compared with those in literature [28], [33] and [2] under 30 attributes. Figure 8 shows that the ciphertext storage space also increases with the number of attributes and files. Obviously, our scheme saves more storage space, and with the increase in the number of files, our scheme shows its storage superiority.

According to the aforementioned conclusions, compared with the present [28], [33] and [2] scheme, our scheme is obviously more efficient in terms of calculation and storage cost. Our scheme makes full use of the advantages of hierarchical access tree structure and provides more convenient services and data acquisition efficiency for medical client users. Our scheme is also suitable for lightweight entities (i.e. sensor nodes and mobile terminals).

TABLE 3. Functions comparison of ABE scheme.

scheme	type	Access structure	Multi-authority	Integrity verification	Resist collusion	Policy hidden	Hide GID
[2]	CP-ABE	Access tree	×	×	×	×	×
[28]	CP-ABE	LSSS	√	×	√	√	√
[33]	CP-ABE	And gate	√	×	√	√	√
[37]	HABS	Hierarchical access tree	√	×	×	×	×
[21]	FH-CP-ABE	Hierarchical access tree	×	×	×	×	×
Ours	FH-CP-ABE	Hierarchical access tree	√	√	√	√	√

TABLE 4. The comparison of computation cost.

Component	[2]	[28]	[33]	Ours
Encryption	$[2(\lvert A_{C_1}\rvert)+\cdots+\lvert A_{C_l}\rvert)+l]e_0+le_T$	$[3(\lvert A_{C_1}\rvert)+\cdots+\lvert A_{C_l}\rvert)+l]e_0+3le_T$	$[2(\lvert A_{C_1}\rvert)+\cdots+\lvert A_{C_l}\rvert)+l]e_0+le_T$	$2(\lvert A_{C_1}\rvert+l)e_0+(2k'\lvert A_T\rvert+l)e_T$
Decryption	$l(2\lvert A_u\rvert+1)P_e+[2(\lvert S_1\rvert+\cdots+\lvert S_l\rvert)+2l]e_T$	$l(5\lvert A_u\rvert+1]P_e+[5(\lvert S_1\rvert+\cdots+\lvert S_l\rvert)+l]e_T$	$l(2\lvert A_u\rvert+3]P_e+[2(\lvert S_1\rvert+\cdots+\lvert S_l\rvert)+3l]e_T$	$2(\lvert A_u\rvert+1]P_e+[2(\lvert S_1\rvert+(k'\lvert A_T\rvert+2l)]e_T$
KeyGen	$(2\lvert A_u\rvert+2)e_0$	$(4\lvert A_u\rvert+1)e_0$	$(2\lvert A_u\rvert+8)e_0$	$(4\lvert A_u\rvert+2)e_0$

TABLE 5. The comparison of storage cost.

Component	[2]	[28]	[33]	Ours
PK	$3L_{G_0}+L_{G_T}$	$3L_{G_0}+L_{G_T}$	$2L_{G_0}+L_{G_T}$	$L_{G_0}+L_{G_T}$
MSK	$L_{Z_P}+L_{G_0}$	$3L_{Z_P}$	$2L_{Z_P}+L_{G_0}$	$L_{Z_P}+L_{G_0}$
SK	$2(\lvert A_u\rvert+1]L_{G_0}$	$(4\lvert A_u\rvert+1]L_{G_0}$	$2(\lvert A_u\rvert+1]L_{G_0}$	$2(2\lvert A_u\rvert+1]L_{G_0}$
CT	$[2(\lvert A_{C_1}\rvert)+\cdots+\lvert A_{C_l}\rvert)+l]L_{G_0}+lL_{G_T}$	$[4(\lvert A_{C_1}\rvert)+\cdots+\lvert A_{C_l}\rvert)+l]L_{G_0}+3lL_{G_T}$	$[2(\lvert A_{C_1}\rvert)+\cdots+\lvert A_{C_l}\rvert)+l]L_{G_0}+lL_{G_T}$	$[2(\lvert A_{C_1}\rvert+l)]L_{G_0}+(k'\lvert A_T\rvert+l)L_{G_T}$

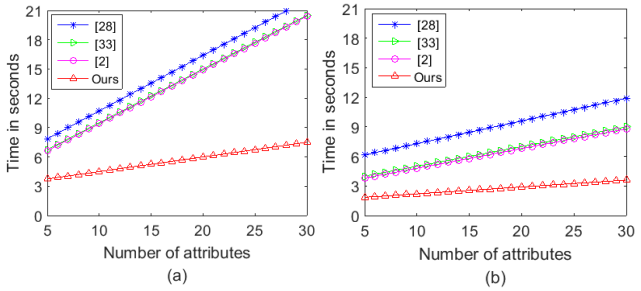

FIGURE 6. (a) Comparison of the encryption time cost for four files. (b) Comparison of the decryption time cost for four files.

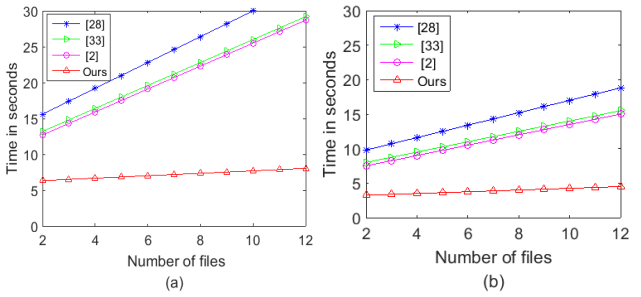

FIGURE 7. (a) Comparison of the encryption cost under 30 attributes. (b) Comparison of the decryption cost under 30 attributes.

In order to better illustrate the usefulness of hierarchical access policies compared with "flat" access policies, Figure 9. (a) and (b) show the comparisons of encryption time and decryption time in [2] and our scheme. In the experiment, the numbers of attributes related with ciphertexts are $|A_{C_1}| = 30, |A_{C_2}| = 27, \cdots, |A_{C_6}| = 15$, respectively. In the case of the current sets, the efficiency of the scheme

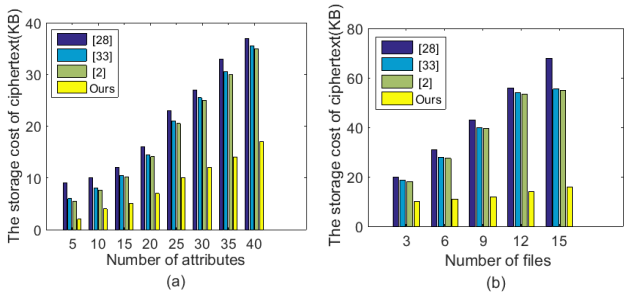

FIGURE 8. (a) Storage cost comparison of four ciphertext files. (b) Storage cost comparison of multiple ciphertext files.

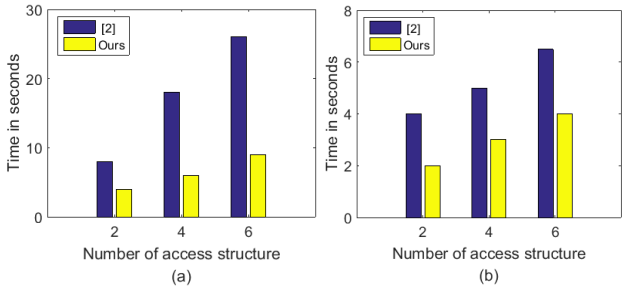

FIGURE 9. (a) Comparison of the encryption time cost. (b) Comparison of the decryption time cost.

is improved as follows: encryption phase 43.75%, 65.71%, 65.38%; decryption phase 50%, 40%, 38.46%.

VII. CONCLUSION AND FUTURE WORK

Based on the application of attribute encryption in modern medical system, this study proposes an anonymous electronic health record-sharing scheme based on decentralized hierarchical ABE in cloud environment. Multi-authority is adopted to realize EHRs sharing sufficiently. The use of hierarchical

access tree structure not only provides fine-grained access control but also improves encryption efficiency by encrypting multiple files at a time and saves storage space greatly. Next, we considered user privacy from two aspects. On the one hand, the hidden access policy improves the confidentiality of shared EHRs. Thus, intermediate user can obtain nothing about user attributes and policy from the access structure. On the other hand, the use of anonymous interaction generates private keys for users. Although multiple AAs own the master key and distribute private keys for user, it cannot decrypt the ciphertext of the user to avoid the key escrow attack of dishonest attribute authority. Moreover, the proposed scheme is proven secured under DBDH assumption. In order to provide better decentralized properties, we will combine blockchain technology in our future work.

APPENDIX

Theorem 1: Suppose DBDH assumption holds. Then no polynomial adversary can selectively break the proposed N-authorities this scheme.

Proof: Suppose that the adversary $\mathcal{A}$ has non-negligible advantage $\varepsilon = Adv\mathcal{A}$ in the selective security game against our construction. Then, we construct a simulator $\Re$ that can distinguish a DBDH tuple from a random tuple with advantage $\varepsilon/2$. Let $e : G_0 \times G_0 = G_T$ is an efficiently computable bilinear map, where G_0 has prime order p with generator g. Firstly, the DBDH challenger randomly selects the following parameters: $a, b, c \in Z_p, \mu \in \{0, 1\}$ and $e(g, g)^{abc} \in G_T$. If $\mu = 0$, challenger sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ to $\Re$, otherwise, it sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^d)$ to $\Re$, where $a, b, c, d \in Z_p$. The simulator $\Re$ now plays the role of challenger in the security game. In order to make the description clearly, only one file is encrypted. Challenger $\mathcal{C}$ and adversary $\mathcal{A}$ play the following games:

Initialization: The adversary $\mathcal{A}$ submits the access structures Γ_0^*, Γ_1^* he wishes to challenge upon and a list of corrupted authorities C_A to algorithm $\Re$, where $|C_A| < N$.

Setup: $\Re$ randomly chooses $A_j^* \in \{A_1, A_2, \cdots, A_N\}$.

(1) For $A_j \in C_A$, $\Re$ randomly chooses a number $v_i \in Z_p$, $w_{ji} \in Z_p$, sets $h_{ji} = g^{w_{ji}}$, for $a_{j,i} \in \tilde{A}_j$,then sends $\langle e(g, g)^{v_j}, \{h_{ji} = g^{w_{ji}}\}_{\forall i \in \tilde{A}_j}\rangle, \langle g^{v_j}, \{w_{ji}\}_{\forall i \in \tilde{A}_j}\rangle$ to $\mathcal{A}$.

(2) For $A_j \notin C_A$, $\Re$ randomly chooses $v_i, w_{ji} \in Z_p$, sets $h_{ji} = g^{w_{ji}} = g^b = B$ for and $\Re$ randomly chooses $w'_{ji} \in Z_p$, sets $h_{ji} = g^{w'_{ji}+a} = Ag^{w'_{ji}}$, for $a_{j,i} \notin \Gamma^*$. If $A_j \neq A_j^*$, $e(g, g)^{\alpha_j} = e(g, g)^{v_j+ab} = e(g, g)^{v_j} e(g, g)^{ab}$. For honest authority A_j, $\Re$ sends PK_j to adversary $\mathcal{A}$.

QueryPhase 1: In this phase, $\mathcal{A}$ can query the secret key by submitting an attribute set $\omega_i = \{\eta_i \in \Gamma\}(\omega_i \notin \Gamma_\theta^*, \theta \in \{0, 1\})$ to $\Re$.

(1) $A_j \in C_A$, $\Re$ randomly choose $r_j \in Z_p, r_i \in Z_p^*, x_u \in Z_p$ and uses $\langle g^{v_i}, w_{ji}\rangle$ to compute secret keys for corresponding attribute sets.

(2) For $A_j \notin C_A$, $\Re$ randomly picks a number $r_j \in Z_p$, $x_u \in Z_p$, and uses $\langle g^{v_i}, w_{ji}, r_u\rangle$ to computes $k_{1i} = g^{v_j} h_{ji}^{r_j/x_u}$, $k_{2i} = g^{r_j} \cdot H(i)^{r_i/x_u(r_u+w_{ji})}$, $k_{3i} = h_{ji}^{r_i/x_u(r_u+w_{ji})}$ for $a_{j,i} \in \Gamma_\theta^*$. Then $k_{1i} = g^{v_j} h_{ji}^{r_j/x_u}$, $k_{2i} = g^{r_j} \cdot H(i)^{r_i/x_u(r_u+w'_{ji}+a)}$, $k_{3i} = h_{ji}^{r_i/x_u(r_u+w'_{ji}+a)}$, for $a_{j,i} \notin \Gamma_\theta^*$. If $A_j \neq A_j^*$, $\Re$ randomly picks $r'_j \in Z_p, x'_u \in Z_p$, sets $m'_j = r'_j/x'_u, m_j = r_j/x_u = r'_j/x'_u - a, v_j = v'_j + ab$. It can obtain $k_{1i} = g^{v_j} h_{ji}^{r_j/x_u} = g^{v_j} g^{w_{ji} m_j} = g^{v'_j+ab} g^{b(m'_j-a)} = g^{(v'_j+m'_j b)}$. Then, for each attribute $\eta_i \in \omega_i$, $\Re$ needs to randomly choose $r_j \in Z_p$, $r_i \in Z_p^*$, sets $r_j = r'_j - a$. It constructs the remaining secret key as follows:

$$k_{2i} = g^{(r'_j-a)} \cdot H(i)^{r_i/x_u(r_u+w_{ji})} = g^{r'_j}/A \cdot H(i)^{r_i/x_u(r_u+w_{ji})},$$

$$k_{3i} = h_{ji}^{r_i/x_u(r_u+w_{ji})} = g^{w_{ji} r_i/x_u(r_u+w_{ji})} = B^{r_i/x_u(r_u+w_{ji})}$$

At last, $\Re$ sends secret key to $\mathcal{A}$.

Challenge: Adversary $\mathcal{A}$ submits two equal-length messages m_0 and m_1 to the simulator $\Re$. $\Re$ flips a random coin $\hat{\mu} \in \{0, 1\}$ and computes CT^* as: $C'_k = g^{s_k} = g^c = C$, $\tilde{C}_k = m_{\hat{\mu}} \cdot e(g, g)^{v_j s_k} = m_{\hat{\mu}} \cdot e(g, g)^{v_j c} = m_{\hat{\mu}} \cdot Ze(g, g)^{v'_j c}$.

Finally, $\Re$ sends CT^* to $\mathcal{A}$.

QueryPhase 2: Same as the QueryPhase 1.

Guess: Finally, $\mathcal{A}$ guesses $\hat{\mu}', \hat{\mu}$. If $\hat{\mu}' = \hat{\mu}$, $\Re$ output 0, that is $Z = e(g, g)^{abc}$;Otherwise output 1, Z is the random group element in G_T.

Probability Analysis: If $Z = e(g, g)^{abc}$, then CT is the correct ciphertext, so there is

$$\Pr[B(g, g^a, g^b, g^c, Z = e(g, g)^{abc}) = 0] = Adv(\mathcal{A}) + 1/2.$$

If Z is just random elements of the G_T, the plaintext is completely hidden for $\mathcal{A}$, it is impossible for $\mathcal{A}$ to obtain any information about μ from the ciphertext, therefore

$$\Pr[C(g, g^a, g^b, g^c, Z = e(g, g)^d) = 0] = 1/2$$

To sum up, the advantage of $\Re$ in solving DBDH problem is $Adv(\mathcal{A})/2$, if $Adv(\mathcal{A})$ cannot be ignored, then the advantage of $\Re$ in solving DBDH problem can't be neglected.

ACKNOWLEDGMENT

The authors wish to thank the anonymous referees for their patience in reading this manuscript and their invaluable comments and suggestions.

REFERENCES

[1] Y. Albagory, "An efficient WBAN aggregator switched-beam technique for isolated and quarantined patients," *Int. J. Electron. Commun.*, vol. 123, no. 153322, pp. 1434–8411, Aug. 2020.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in *Proc. IEEE Symp. Secur. Privacy*, pp. 321–334, May 2007.

[3] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.

[4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Public Key Cryptogr.-PKC*, 2011, pp. 53–70.

[5] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. 35th Int. Colloq.* Berlin, Germany: Springer, Reykjavik, Iceland, Jul. 2008, pp. 579–591.

[6] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Proc. 10th Int. Workshop Inf. Secur. Appl. (WISA)* (Lecture Notes in Computer Science), vol. 5932, H. Y. Youm and M. Yung, Eds. Berlin, Germany: Springer, 2009, pp. 309–323.
[7] C.-K. Chu, W.-T. Zhu, J. Han, J. K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervas. Comput.*, vol. 12, no. 4, pp. 50–57, Oct. 2013.
[8] T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. Liu, "TIMER: Secure and reliable cloud storage against data re-outsourcing," in *Proc. 10th Int. Conf. Inf. Secur. Pract. Exper.*, vol. 8434, May 2014, pp. 346–358.
[9] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud based revocable identity-based proxy re-encryption scheme for public clouds data sharing," *Comput. Secur.*, vol. 8712, pp. 257–272, Sep. 2014.
[10] T. H. Yuen, Y. Zhang, S. Yiu, J. K. Liu, "Identity-based encryption with post-challenge auxiliary inputs for secure cloud applications and sensor networks," *Comput. Secur.*, vol. 8712, pp. 130–147, Sep. 2014.
[11] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X. Phuong, and Q. Xie, "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1667–1680, Oct. 2014.
[12] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Proc. 10th Int. Workshop Inf. Secur. Appl.*, Aug. 2009, pp. 309–323.
[13] X. Xie, H. Ma, J. Li, and X. Chen, "An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing," *J. Universal Comput. Sci.*, vol. 19, no. 16, pp. 2349–2367, Oct. 2013.
[14] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 763–771, May 2014.
[15] A. Balu and K. Kuppusamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," *Inf. Sci.*, vol. 276, pp. 354–362, Aug. 2014.
[16] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *Int. J. Netw. Secur.*, vol. 16, no. 6, pp. 437–443, Nov. 2014.
[17] Y. Chen, Z. L. Jiang, S. M. Yiu, J. K. Liu, M. H. Au, and X. Wang, "Fully secure ciphertext-policy attribute based encryption with security mediator," in *Proc. 16th Int. Conf. Inf. Commun. Secur.*, vol. 8958, Dec. 2014, pp. 274–289.
[18] Y. Yang, J. K. Liu, K. Liang, K. R. Choo, and J. Zhou, "Extended proxy-assisted approach: Achieving revocable fine-grained encryption of cloud data," *Comput. Secur.*, vol. 9327, pp. 146–166, Sep. 2015.
[19] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, and A. Yang, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Gener. Comput. Syst.*, vol. 52, pp. 95–108, Nov. 2015.
[20] X. Liu, Y. Xia, W. Yang, and F. Yang, "Secure and efficient Querying over personal health records in cloud computing," *Neurocomputing*, vol. 274, pp. 99–105, Jan. 2018, doi: 10.1016/j.neucom.2016.06.100.
[21] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016, doi: 10.1109/TIFS.2016.2523941.
[22] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc Int. Conf. Theory Appl. Cryptograph*, 2005, pp. 457–473.
[23] V. Goyal, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.
[24] M. Chase, "Multi-authority attribute based encryption," in *Proc. 4th Theory Cryptogr. Conf.*, 2007, pp. 515–534.
[25] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi-authority attribute based encryption without a central authority," in *Proc. 9th Int. Conf. Cryptol. India*, 2008, pp. 426–436.
[26] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 121–130.
[27] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Adv. Cryptol.*, 2011, pp. 568–588.
[28] P. Liang, L. Zhang, L. Kang, and J. Ren, "Privacy-preserving decentralized ABE for secure sharing of personal health records in cloud storage," *J. Inf. Secur. Appl.*, vol. 47, pp. 258–266, Dec. 2019.
[29] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 35–45, Jan. 2016.
[30] R. Xu, and B. Lang, "A cp-abe scheme with hidden policy and its application in cloud computing," *Int. J. Cloud Comput.*, vol. 4, no. 4, pp. 279–298, 2015.
[31] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," *Soft Comput.*, vol. 22, no. 1, pp. 243–251, 2016, doi: 10.1007/s00500-016-2330-8.
[32] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT," *Comput. Netw.*, vol. 133, pp. 141–156, Dec. 2018.
[33] H. Qian, J. Li, and Y. Zhang, *Privacy-Preserving Decentralized Ciphertext-Policy Attribute-Based Encryption With Fully Hidden Access Structure*. Cham, Switzerland: Springer, 2013, pp. 363–372.
[34] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in Cryptology*. Berlin, Germany: Springer, Dec. 2002, pp. 548–566.
[35] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 735–737.
[36] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 743–754, Apr. 2012, doi: 10.1109/tifs.2011.2172209.
[37] J. Shen, D. Liu, Q. Liu, X. Sun, and Y. Zhang, "Secure authentication in cloud big data with hierarchical attribute authorization structure," *IEEE Trans. Big Data*, early access, May 17, 2017, doi: 10.1109/TBDATA.2017.2705048.
[38] A. Kate, G. Zaverucha, and I. Goldberg, "Pairing-based onion routing," in *Proc. Int. Workshop Privacy Enhancing Technol.*, Berlin, Heidelberg: Springer, 2007, pp. 95–112.
[39] V. Lynn. (Jan. 2006). *PBC Libarary*. [Online]. Available: http://cryptostanford.edu/pbc

XUEYAN LIU received the B.S. and M.S. degrees from Northwest Normal University, in 2001 and 2004, respectively, and the Ph.D. degree from the Lanzhou University of Technology, in 2016. She has been working in Northwest Normal University, since July 2001, and she is currently an Associate Professor and a Master Tutor. She has published some wonderful articles and she is a reviewer of some journals. She also hosts and participated in several National Natural Science Foundation projects. Her main research interests include cryptographic, information security, and the Internet of Things security.

XIAOTAO YANG is currently pursuing the master's degree with the School of Mathematics and Statistics, Northwest Normal University. Her research interests include attribute cryptography and information security.

YUKUN LUO graduated in mathematics and applied mathematics from Northwest Normal University, in 1996, where she is currently pursuing the master's degree. Her main research directions are lattice cryptography and cloud storage.

LI WANG is currently pursuing the master's degree with the School of Mathematics and Statistics, Northwest Normal University. Her research interests include the mobile crowdsensing and information security.

QIANG ZHANG graduated from Northwest Normal University, Lanzhou, China, in July 2001. He received the master's degree in computer application from Northwest Normal University, in 2004, the Ph.D. degree in management science and engineering from the Xi'an University of Technology, Xi'an, China, in 2011, and the Postdoctoral degree in electrical engineering from Xi'an Jiaotong University, Xi'an, in 2014. He is currently an Associate Professor and a Master's Tutor with the Institute of Computer Science and Engineering, Northwest Normal University. He presided over the completion of the project of the National Natural Science Foundation of China, the project of the Ministry of Education, and more than 20 other projects.