Ciphertext-Policy Hierarchical Attribute-Based Encryption Against Key-Delegation Abuse for IoT-Connected Healthcare System

XI CHEN\(^1\), YUN LIU\(^1\), HAN-CHIEH CHAO\(^2\), (Senior Member, IEEE), AND YONG LI\(^1\)

\(^1\)Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China
\(^2\)Department of Electrical Engineering, National Dong Hwa University, Hualien 974, Taiwan
Corresponding authors: Yun Liu (liuyun@bjtu.edu.cn) and Yong Li (liyong@bjtu.edu.cn)

ABSTRACT With the maturity of technologies such as Internet of Things (IoT) and Cloud Computing, more and more medical institutions share patient’s medical data in the IoT-connected healthcare system, whereas it also brings hidden dangers to data security and privacy protection. The application of Hierarchical Attribute-Based Encryption (HABE) in IoT-connected healthcare system helps to solve the problem of data sharing with a large number of users. Delegation is a function of HABE, which can realize the transfer of hierarchical user access rights, effectively reducing the workload of the trusted authority. However, when “delegation” is used to generate decryption keys for users who do not comply with the access structure, the “key-delegation abuse” problem arises, which can seriously damage the privacy of patient’s data. Nevertheless, it has not attracted much attention in previous research. This paper proposes the problem of key-delegation abuse in Ciphertext-Policy Hierarchical Attribute-Based Encryption (CP-HABE). After the cryptanalysis of typical schemes, we conclude two reasons for key-delegation abuse in CP-HABE, which are randomizing the original key elements or keeping parts of them in a new decryption key. This paper proposes a new mechanism specifically by using directed graph and construct a CP-HABE scheme against key-delegation abuse (CP-HABE-AKDA). Aiming at the scenario of key leaking, we further present the traceable CP-HABE-AKDA solution for the IoT-connected healthcare system, which has the additional function of tracking and verifying the identity of key leaker.

INDEX TERMS Internet of Things, hierarchical attribute-based encryption, key-delegation abuse, ciphertext-policy, white box traceability.

I. INTRODUCTION

As one of the most potent communication paradigms, the Internet of Things (IoT) connects many objects (such as sensors, vehicles, houses and devices) to the Internet together. Since Cloud Computing [1] and Edge Computing [2] provide critical technical support for the IoT application layer, IoT can transmit real-time information to the target users and achieve massive storage at the same time. With the proliferation of concepts and products such as smart city, intelligent transportation, smart home, Wise Information Technology of 120, etc., the application scope of IoT has penetrated all aspects of people’s lives. IoT has become a key component in healthcare applications. For example, wearable or embedded sensors can be connected to the subject’s body for monitoring physiological status and collecting physiological symptoms. The collected data, such as body temperature, blood pressure, heartbeat, etc. constitute the subject’s electronic health record (EHR), which is one of the core components in IoT-connected healthcare system. According to EHR, medical personnel can provide patients with timely treatment of acute diseases and accurate prevention of chronic diseases, thereby IoT-connected healthcare system plays a crucial role in promoting the improvement of medical and healthcare services.

With the rapid development of IoT-connected healthcare system, data security and privacy issues have raised concerns about the development of smart healthcare. For example, a large amount of user information gathered from IoT devices is uploaded to the cloud in real time. Attackers illegally obtain massive data to design and develop healthcare applications [3]. The adversary also uses sensors from wearable devices to track patients without their consent.
The promulgation of the EU’s General Data Protection Regulation (GDPR) has made personal data protection even more stringent. The laws of many countries treat individuals’ privacy as a fundamental right. The U.S. health insurance privacy and accountability act (HIPAA) does not explicitly limit the use of wearables to collect physiological data, but it also requires compliance with federal regulations, which prohibit the disclosure of medical information without a patient’s consent. The public’s concern about data privacy and the enactment of laws related to privacy protection urge researchers to design more reasonable privacy conservation schemes and information confidentiality approaches.

Considering the privacy sensitivity of EHR, encryption and authorization are the leading measures for data owners to ensure privacy and enhance personal data control. Attribute-based encryption (ABE) is a critical advanced encryption mechanism for implementing encryption, authorized access and sharing information with multiple users, which provides data confidentiality and fine-grained access control for EHR. The delegation feature in ABE schemes [4]–[6], which allows delegating to a subset of user attribute sets, and provides an ideal solution to the dynamically expanding problem brought by large-scale users. Whereas, the application of ABE in IoT-connected healthcare system remains practical challenges. For example, it is expected to have distinct access privileges for authorized users sharing EHR at different levels, such as physician-in-charge, attending doctor and medical doctor. Ciphertext-policy hierarchical attribute-based encryption (CP-HABE) is proposed as a prospective primitive for solving the above problem, which can achieve fine-grained access control in a hierarchy.

### A. MOTIVATIONS

In order to deploy the CP-HABE scheme in IoT-connected healthcare system, besides implementing hierarchical access control on encrypted EHR, it is also essential to reduce the heavy key management workload of root authority. Some CP-HABE solutions [7], [8] support the key delegation from higher-level to lower-level, thus reducing the burden of root authorities. However, when delegation is leveraged to generate decryption keys for users who do not conform to the access structure, as is shown in Fig.1, it will lead to the abuse of key-delegation, which would not only destroy the access policy but also put the private data among the security risk of illegal access. In 2018, Jiang et al. [9] raised the issue of key-delegation abuse in CP-ABE and proposed a new scheme to resist key-delegation abuse in fog computing, but they did not give a definition of the problem, nor did they analyze its causes. CP-ABE and CP-HABE have distinct approaches of key-delegation abuse, but both of them pose a threat on the data access security; thus, the strategies to deal with the abuse of key-delegation are not the same. Furthermore, there is still a lack of relevant research and practical solutions to the key-delegation abuse in CP-HABE systems. Therefore, for the privacy protection in the IoT-connected healthcare system, especially based on CP-HABE, it is an essential issue that how to make use of the delegation to provide the lightweight key management in the CP-HABE scheme without causing key-delegation abuse.

Secondly, the massive data in IoT-connected healthcare system owns high commercial value. Data users are possible to disclose the key for the sake of benefits. What’s more, different users with specific attributes in the hierarchy can decrypt the ciphertext containing the same access structure. That may give rise to the users’ disclosing their keys without worrying about identities exposure. Hence, the CP-HABE solution applied in IoT-connected healthcare system is exceedingly required to have effective methods of tracking down the key leaker.

### B. CONTRIBUTIONS

Aiming at the above problems, we construct a ciphertext-policy hierarchical attribute-based encryption scheme against key-delegation abuse (CP-HABE-AKDA). Taking the IoT-connected healthcare system as the application case, we present a traceable CP-HABE-AKDA scheme for the scenario where a user discloses the private key. The main advantages of this paper are summarized as follows:

- We first point out that key-delegation in the CP-HABE scheme will lead to key-delegation abuse, which would destroy the access policy. We cryptanalyze Asim et al.’s dynamic CP-HABE [7] scheme and discover the vulnerability in the key delegation algorithm. The domain authority can generate decryption keys for users who do not comply with the access policy, resulting in unauthorized data access. We also conclude that when the higher-level domain authority delegates the key to the lower-level domain authority, randomizing the higher-level key elements to generate a new key, or retaining its decryption parts in the new lower-level key, are two ways that may cause the abuse of key-delegation in CP-HABE.

![FIGURE 1. A general architecture of IoT-connected healthcare system based on CP-HABE.](image-url)
• We achieve the delegation between hierarchical domain authorities while preventing key-delegation abuse. The proposed CP-HABE-AKDA scheme uses the integration of the directed graph and CP-HABE to realize the delegation of domain authorities. We embed a level key parameter into the user attribute private key, which is to prevent the higher-level domain authority from using the delegated new level key to forge a user attribute private key for unauthorized users. When applied to IoT-connected healthcare systems, the solution will significantly reduce the amount of workload for the root authority, which is associated with multiple IoT devices and users, achieving the scalability of the system.

• We further study the strategy of tracking malicious user’ identity under the key leak scenario and propose a traceable CP-HABE-AKDA solution for IoT-connected healthcare system. The EHR user identity is bound to the attribute set and embedded in the user attribute private key for becoming a necessary part of decryption. If a key is compromised, the identity of the key holder can verifiably be traced.

C. OUTLINE
The paper is organized as follows: In Section 2, we discuss related works. In Section 3, we discuss the key-delegation abuse problem in CP-HABE schemes. Section 4 provides some background definitions and the system description. In Section 5, the CP-HABE-AKDA scheme is presented, and the security proof is given in Section 6. In Section 7, we further proposed a traceable CP-HABE-AKDA scheme for IoT-connected healthcare system, and the security and traceability proof is given in Section 8. In Section 9, the comparisons of theoretical performance with some related works are provided. Finally, in Section 10, we conclude with some discussions and future work.

II. RELATED WORK
A. ATTRIBUTE-BASED ENCRYPTION
Attribute-Based Encryption (ABE) is a one-to-many advanced encryption mechanism. Delegation is a property of ABE, which can not only improve the efficiency of key management but also reduce the load of key issuing agencies. The delegation feature provides ABE with the necessary support for data sharing scenarios with a large number of users, such as cloud storage.

In 2005, Sahai and Waters [10] first proposed an ABE scheme as a fuzzy identity encryption application, in which data owners do not need to know the identity. The user’s decryption key is associated with a set of attributes, and the user can decrypt the ciphertext only if there is a match between the decryption key and the ciphertext. In 2006, Goyal et al. [4] embedded the access structure into the user key and used the attribute set to describe the ciphertext, proposing an ABE (KP-ABE) scheme that supports the key strategy of any monotonic access structure. The solution also first proposed the delegation of decryption keys. When Y structure is subject to X structure, the user belonging to the access structure X can delegate the decryption key to the user whose access structure is Y.

In 2007, Ostrovsky et al. [11] further studied and constructed a KP-ABE scheme that allowed non-monotonic access structures to achieve a more flexible access strategy. Bethencourt et al. [5] proposed the ciphertext policy ABE (CP-ABE) scheme, in which the ciphertext and key roles are opposite to the KP-ABE scheme. The message is encrypted with a tree access policy, and the corresponding decryption key is associated with a set of attributes. This key can be used to decrypt the ciphertext as long as the attribute set satisfies the tree access policy, which is related to given ciphertext. The CP-ABE scheme also supports delegation, when the user can generate a key for a subset of its attribute set to implement delegation for the decryption key.

Cheung et al. [12] improved the security proof in Bethencourt et al. [5], using the AND-gate positive and negative attributes as the access structure, and proved for the first time that the selected ciphertext is secure under the assumption of the Decision Bilinear Diffie-Hellman (DBDH). However, this scheme is not richly expressive, and the ciphertext and key size increase linearly with the number of attributes. In 2008, Goyal et al. [13] proposed an improved version of CP-ABE, supporting access tree structures with bounded thresholds, and proved the decision-making bilinear Diffie-Hellman hypothesis under the standard model. The problem with this scheme is that the depth of the tree needs to be set during the initialization phase, and the user may be limited to using only access trees whose depth is less than the preset value. In 2011, based on the Linear Secret Sharing Scheme (LSSS) matrix, Waters [6] proposed a scheme with attribute based access control policies that support key delegation, but the size of ciphertext is linearly increasing with encryption and decryption overhead. In 2017, Li et al. [14] introduced an ordered binary decision graph (OBDD) as a new access structure to improve CP-ABE, a non-monotonic access structure that supports “AND, NOT, OR” relationship. In 2018, the key-delegation abuse in CP-ABE was proposed for in Jiang et al. literature [9] and the new schemes of provable security and traceable protection against key-delegation abuse were proposed in the fog computing scenario. However, malicious users can still build decryption devices or online black boxes to illegally decrypt other people’s ciphers, with little risk of getting caught. Therefore, Qiao et al. [15] proposed the first practical CP-ABE scheme with black box traceability and extensibility. More recently, Yu et al. [16] propose a traceable and undeniable CP-ABE scheme, which can judge the traced malicious user is innocent or not, by introducing a public auditor. When CP-ABE is applied in the information centric Internet of things (ICoT) to realize the flexible access control of IoT data, the data publisher’s retrieval of attributes on the centralized server will lead to high communication overhead. To address this issue, in 2017, Li et al. [17] proposed a distributed publisher driven secure data sharing for ICoT,
which enables only authorized users to retrieve IoT data from distributed cache.

B. HIERARCHICAL ATTRIBUTE-BASED ENCRYPTION

When a large number of users present a hierarchical structure, the CP-ABE scheme cannot support a complete delegation mechanism and an efficient, extensible revocation mechanism. To solve this problem, in 2011, Li et al. [18] proposed the concept of HABE, which classifies common attributes into a tree hierarchy, and each node on the tree is associated with an attribute. The parent node of the tree structure can derive the child node key to implement delegation, and vice versa. The user can decrypt the ciphertext only if the user attribute covers the number of attributes in the ciphertext and is not lower than a certain threshold. However, due to the use of thresholds, this solution does not provide fine-grained access control. Wang et al. [19] proposed a fully authorized HABE by combining identity-based hierarchical attribute encryption (HIBE) with CP-ABE and supported an extensible revocation mechanism. In 2011, Wang et al. [20] implemented key-delegation between attribute authorities to provide confidentiality for shared data stored on untrusted cloud servers, and more suitable for enterprise outsourcing data in a cloud environment. However, since different domain masters can maintain the same attributes, this leads to more complexity, and the scheme does not support composite attributes.

In 2012, Wan et al. [21] proposed a hierarchical attribute set-based encryption scheme (HASBE) that supports scalability, fine-grained access control, and user revocation. In this scenario, the data owner or data consumer is online only when needed, and does not have to be online from time to time, but the attribute authority should always be online. Asim et al. [7] combined encryption technology with trust management to define a dynamic HABE (D-HABE) scheme, which does not require encryption to bind attributes in access control policies to specific domain authorities. The scheme makes new users who join the authority domain of the virtual organization can decrypt the original information without re-encrypting them; however, the flexible key construction of the solution can lead to abuse of key delegation. In 2014, Liu et al. [22] proposed a CP-HABE scheme for fine-grained access control and proved that the scheme is secure under the assumption of q-parallel bilinear Diffie-Hellman exponent. Deng et al. [8] proposed a CP-HABE scheme with short ciphertext through dual-system encryption technology [23]–[25], in which attributes are organized in a matrix form. This feature enables the CP-HABE to carry a large number of users from different organizations with a delegation key. In 2016, Wang et al. [26] designed an efficient file-level ABE scheme for file sharing with hierarchical structure in cloud computing, which reduced ciphertext storage complexity and saved encryption time cost. In 2019, Li et al. [27] pointed out that Deng et al. scheme [8] did not consider leakage attacks. In order to solve this problem, the paper proposes a formal definition and security model of CLR layered attribute encryption (CLR-HABE) and constructs a CLR-HABE scheme against leak detection.

When applied in IoT related scenarios, the delegation of HABE is conducive to decreasing the workload of IoT devices in resource-constrained environments. In 2018, Huang et al. [28] proposed a fine-grained data access control scheme for restricted IoT devices and cloud computing based on HABE, which reduced the key management workload of trusted authority through employing an appropriate delegation mechanism in HABE. In 2018, Wang and Gao [29] proposed a regulation scheme with access control of identities for regulating Bitcoin transactions. The scheme is based on CP-HABE, in which users’ identities are encrypted by using access policy and are contained in their transaction. Attributes of high level are assigned to regulation nodes, which have more priorities than ordinary users that presented by attributes of low level. In 2019, Sun et al. [30] proposed a fine-grained ranked multi-keyword search scheme over hierarchical data for the IoT-oriented healthcare system by leveraging CP-HABE and ranked multi-keyword search (RMKS) technologies. In 2019, Tan et al. [31] proposed a lightweight KP-ABE scheme to support IoT, and expanded it to a KP-HABE scheme for role delegation of healthcare systems connected with IoT. However, none of the above schemes mentions the problem of key-delegation abuse in HABE.

III. KEY-DELEGATION ABUSE IN CP-HABE

This section will analyze the algorithm of the CP-HABE schemes owning the property of delegation, explain the problem of “key-delegation abuse” and summarize the causes of the problem. For simplicity, the notations employed throughout this paper are listed in Table 1.

Firstly, the definition of key-delegation abuse in CP-HABE is given, that is, the domain authority or the user at level $i, 1 < i < L$, can generate decryption key by delegation for the user at level $h, 1 < i < h < L$, who does not conform to the access policy. Such behavior that destroying the access policy is called key-delegation abuse in CP-HABE. The following example will illustrate how key-delegation abuse could form and break access policy. For the sake of generality, the dynamic CP-HABE scheme proposed by Asim et al. [7] is considered.

A. REVIEW OF ASIM et al. SCHEME

Briefly introduce the main functions of Asim et al. scheme [7]:

\[
\text{Setup} (\lambda, L) : \text{The root authority RA executes the initialization algorithm to generate parameters for a hierarchy of depth}\ L. \text{The assumption is that at the first level of the hierarchy there are } \Psi \text{ domains. The input security parameters is } \lambda, \text{ and } g \text{ is a random generator of } G_0. \text{ The algorithm randomly selects } \alpha, \beta \in Z_p, A = e(g, g)^{\alpha - \beta}, g_1 = g^{\alpha}, g_2 = g^{\beta}, g_3, h_1, h_2, \ldots, h_L \in G_0, y_1, y_2, \ldots, y_\Psi \in Z_p. \text{ The public and secret parameters are composed of the following steps:}
\]

1. \text{Key Generation:} \text{For a user with attributes } \{a_1, a_2, \ldots, a_n\}, \text{ the key authority RA generates the key with the attributes as input. The key is calculated as:}

\[
\text{Key} = \prod_{a_i \in \text{attributes}} h_i^{a_i} \in G_0.
\]

2. \text{Decryption:} \text{A user can decrypt the ciphertext if the user's attributes satisfy the access policy. The decryption process is:}

\[
\text{Decrypted Data} = \text{Ciphertext} \cdot \text{Key} \in G_0.
\]

3. \text{Access Control:} \text{The decryption is valid if and only if the user's attributes satisfy the access policy, which is determined by the access policy generation function:}

\[
\text{Access Policy} = \prod_{a_i \in \text{attributes}} g_1^{a_i} \in G_0.
\]

The above steps ensure the security and efficiency of the CP-HABE scheme.
components:

- **PK** = (g, g3, h1, h2,..., hL, A)
- **MK** = (g1, g2, 3l, B = \{y1, y2,..., yψ\})

The root authority RA performs this algorithm to generate secret key **SK** for the domain authority **DA** of level i (1 ≤ i ≤ L). The algorithm randomly selects r, yφ ∈ Zp, yψ ∈ B, and generates a key for this domain authority at level i:

**SK**i = \( \left( g^a, g^3 \prod_{i=1}^{L} h_i, g^r, h'_r, \ldots, h''_r, g^{b - 3yφ - yψ}, yφ + yψ, \forall j ∈ Ω_{adm} : g^{3yφ + yψ} H(j)^{yφ + yψ} \right) \)

A domain authority can use its secret key to generate secret keys for domain authorities beneath its level. Specifically, a domain authority key **SK** of a level i can be generated in an incremental manner by the key **SK**i−1 for its parent node in the hierarchy. Let **SK**i−1 be the key of the parent node:

**SK**i−1 = \( \left( g^a, g^3 \prod_{i=1}^{L−1} h_i, g^r, h'_r, \ldots, h''_r, g^{b - 3yφ - yψ}, yφ + yψ, \forall j ∈ Ω_{adm} : g^{3yφ + yψ} H(j)^{yφ + yψ} \right) \)

To generate **SK**i, the domain authority corresponding to the parent node picks randomly rψ ∈ Zp and yφψ ∈ Zp, where

\[ r = r' + r'', y_φ = y_{φ'} + y_{φ''}, \]  
and outputs:

\[ SK_i = \left( a_0, b_1, g^3 \prod_{i=1}^{L} h_i, g^r, a_1, g^{y_{φ'}}, b_{i+1} \right) \]

\[ \cdot h_{i+1}, \ldots, b_i, h'_r, \ldots, h''_r, \]
\[ c_0, c_1, g^{-3yφ} \prod_{j} j^\psi, \forall j ∈ Ω_{adm}' : g^{3yφ} H(j)^{yφ} \]

**AttributeKeyGeneration** (**SK**i, **PK**, ω): The user attribute private key generation algorithm is run by a domain authority, which selects random number x ∈ Zp for each user, and outputs the user attribute private key with attribute set ω at level i (1 ≤ i ≤ L), as follows:

\[ SK_{i, ω} = \left( g^a, g^3 \prod_{i=1}^{L} h_i, g^r, g^{b - 3yφ - x - yψ}, \right) \]
\[ \forall j ∈ Ω_{adm} : D_j = g^{3yφ + x + yψ} H(j)^{yφ} \]

**Encryption** (**PK**, **M**, τ, i): The algorithm encrypt message M ∈ G1 with the inputs of access tree τ and level i, which are related to access policy. Let z be a node of access tree τ, the encryption algorithm selects a random value x ∈ Zp and a polynomial qz (·) for each node z of τ in a top-down manner, starting from the root node R, outputs cipher text as follows:

\[ CT = \left( M \cdot A^x, g^s, g^3 \prod_{i=1}^{L} h_i, \right)^s \]
\[ ∀K, K ∈ τ : C_{att(K)} = g^{6x_0(0)} \]
\[ C'_{att(K)} = H \left( att(K) \right)^{gK(0)} \]

\[ = (C', C_0, C_1, ∀K, K ∈ τ : C_{att(K)}, C'_{att(K)}) \]

**Decryption** (**CT**, **SK**i, ω): First, the decryption algorithm uses a recursive algorithm **DecryptionNode** (**CT**, **SK**i, ω, z) to determine whether the user attribute set ω satisfies the access tree τ; then, the plaintext is recovery.

1. **DecryptionNode** (**CT**, **SK**i, ω, K): Let K is a leaf node on the access tree, j = att (K), each leaf node is associated with a real-valued attribute. If j ∈ ω, then

\[ DecryptionNode(C_T, SK_{i, ω}, K) = e(D_j, C_j) \]
\[ e(D', C_j') \]
\[ = e(g, g)^{(3yφ + x + yψ)qK(0)} \]

If j \not∈ ω, then **Decryption** (**CT**, **SK**i, ω, K) = ⊥.

2. **DecryptionNode** (**CT**, **SK**i, ω, k): When k is an internal node of the access tree, let the nodes z be the children of k, Ωk be an arbitrary set of child nodes. The algorithm output Fz to determine whether the user has enough attributes to satisfy the access policy. If not, the function returns ⊥, otherwise, using polynomial interpolation, the algorithm evaluates the following function:

\[ F_k = \prod_{z ∈ Ω_k} F_z^{Δ_k(0)} \]
\[ ν = \text{index} (z) \]
\[ = \prod_{z ∈ Ω_k} \left( e(g, g)^{(3yφ + x + yψ)q_k(0)} \right)^{Δ_k(0)} \]
\[ = e(g, g)^{(3yφ + x + yψ)q_k(0)} \]
If the user attribute set meets the access policy, continue with



FIGURE 2. Schematic diagram of key-delegation abuse in Asim Scheme.

If the user attribute set meets the access policy, continue with

\[
Z^{(1)} = DecryptNode(CT, SK_{i,DA}, R) = e(g, g)^{(\beta g + x + y)}
\]

\[
Z^{(2)} = e\left(g^{\beta - \beta g}, C_0\right), Z^{(1)} = e(g, g)\beta x
\]

\[
Z^{(3)} = \frac{e\left(g^z, g^\alpha \cdot g^t \cdot \prod_{i=1}^{l-1} h_i\right)}{e(g, g)\beta s} = e(g, g)\beta s
\]

\[
\tilde{C} \cdot \frac{Z^{(2)}}{Z^{(3)}} = M \cdot A^x \cdot e(g, g)^{\beta s} \cdot e(g, g)^{\alpha x} = M
\]

B. CRYPTOANALYSIS OF ASIM et al. SCHEME

As is shown in Fig. 2, in Asim et al. scheme [7], the domain authority DA_{i-1} of level \(i - 1\) can obtain the domain authority key at level \(i\), through the key-delegation, and forge the attribute key SK_{i,DA} for the user with attribute set \(\omega'\) at level \(i\). That will result in the abuse of key-delegation. The process is as follows:

According to the conditions in the scheme that affect the generation of user attribute keys: the user attribute set meeting the access structure and the attribute set belonging to the correct level in the hierarchy, the process of forgery is mainly divided into two parts. Firstly, the key level SK_i is generated, and then the expression of attribute j in attribute set \(\omega'\) is embedded on the basis of S_{i,DA}.

Step 1. According to KeyGen(\(PK, MK\)), the domain authority DA_{i-1} of level \(i - 1\) uses a level key SK_{i-1} to generate a level key SK_i by delegation.

Step 2. DA_{i-1 generates user attribute private key SK_{i,DA} of attribute set \(\omega'\) at level \(i\), according to the algorithm AttributeKeyGeneration(\(SK_i, PK, \omega'\)). DA_{i-1} selects a ran-

\[
SK_{i,DA} = \left(\begin{array}{c}
\left(\frac{a_0 \cdot b_1 \cdot N^{(r-r')}, a_1 \cdot g^{(r-r')}, c_0 g^{-x}}{(1)}
\right.

\forall j \in \Omega_{adm} : D_j = \left(E \cdot g^{x+\gamma o} \right) (F \cdot H(j)\gamma o)
\end{array}\right)
\]

(3)

\[
D' = c_1 \cdot g^{-\gamma o},
\]

(4)

where \(N = g_3 \cdot \prod_{i=1}^{l-1} h_i, E = g_5 y_o + y_o^{-\gamma o}, F = H(j)\gamma o\).

By analyzing the components of attribute key SK_{i,DA} in Asim et al. scheme, we could see that the key parts (1), (2) are related to the level of the user, and the key part (3), (4) and (5) are related to attribute set. Therefore, in order to satisfy the decryption requirement 1, that the user attribute set needs to be satisfied with the access structure, DA_{i-1} could use part(3), (4) and (5) in the attribute private key SK_{i-1,DA}, which satisfying the access structure to construct the key part related to the attribute set in the attribute key SK_{i,DA} at the level i. In order to satisfy the decryption requirement 2, the attribute set needs to belong to the correct level in the hierarchy. The domain authority DA_{i-1} can construct parts (1) and (2) of SK_{i,DA}, by using some parts of level key SK_i generated by hierarchical delegation.

C. CAUSE ANALYSIS

Formally, in the CP-ABE scheme, the key-delegation is completed by generating the key of its attribute set subset; in the CP-HABE scheme, the key-delegation is completed by generating the lower-level user key. However, from an algorithm perspective, the two types of key-delegation are mainly accomplished through the following two methods:

(1) The original key parts are randomized by taking a random number as the exponent to generate a new key and complete the key-delegation. This situation is found in scheme [7], [5]. The algorithm Delegate in reference [5] is taken as an example to explain the key-delegation process: the inputs of Delegate are the key SK of attribute set \(\tilde{S}\) and the other attribute set \(\tilde{S}\), where \(\tilde{S} \subseteq S\). The algorithm selects random numbers \(\tilde{r}\) and \(\tilde{\eta}\), \(\forall k \in \tilde{S}\), for randomizing each part of the key SK = \(\left(D, \forall j \in S : D_j, D'_j\right)\). Based on SK, the algorithm outputs a new key \(\tilde{S}K\) of attribute subset \(\tilde{S}\).

\[
\tilde{S}K = \left(\tilde{D} = D_f \tilde{r}, \forall k \in \tilde{S} : \tilde{D}_k = D_k g^\tilde{r} H(k)\tilde{r}^2, \tilde{D}'_k = D'_k g^\tilde{r}^2\right)
\]

(2) The core parts of the original key, which are responsible for decryption, are kept in a new key and realize the key-delegation. This situation is found in scheme [8], [32]. The algorithm Delegate(\(PK, SK'_S, S\)) are the public key PK, a secret key SK_{SK'} for S', which is a set of attribute vectors
of depth \( k \) and a set \( S \) of attribute vectors of depth \( k + 1 \). It outputs a new secret key \( SK_S \) for \( S \).

\[
SK_S = \left( K_0, K_1, \{ K_{(j,0)}, K_{(j,1)}, K_{(j,k+1)}, \ldots, K_{(j,L)} \}_{|S|+1} \right)
\]

After the observation of \( K_0, K_1, K_{(j,0)}, K_{(j,1)} \), these key elements can be represented by elements of the original key \( SK_{S'} \), such as \( K_0 = K_{0}^{g^W} R_1, K_1 = K_{1}^{g^W} R_1 \), and \( K_{0}^{i}, K_{1}^{i}, K_{(j,0)}^{i}, K_{(j,1)}^{i} \) are the core parts of the decryption algorithm, as shown below:

\[
B = \prod_{\rho(i) \in S} \left( \frac{e\left(C_{(i,0)}, K_1^{i}\right) \cdot e\left(C_{(i,2)}, K_{(j,1)}^{i}\right)}{e\left(C_{(i,1)}, K_{(j,0)}^{i}\right)} \right)^{\alpha_{i}}
\]

\[
M' = e\left(C_{1}, K_{0}^{i}\right) / B
\]

\[
M = C_{0}/M'
\]

Therefore, the core decryption elements \( K_0^{i}, K_1^{i}, K_{(j,0)}^{i}, K_{(j,1)}^{i} \) of the key \( SK_{S'} \) of depth \( k \) can be used to construct the key \( SK_S \) with a depth of \( k + 1 \), and the key-delegation is achieved from the user of depth \( k \) to the user of depth \( k + 1 \).

Since the above two delegation algorithms are both completed based on the secret key held by the delegators themselves. Therefore, when there is no permission or lack of constraints in a CP-HABE scheme, the delegator is possible to generate new keys for users who do not comply with the access policy and perform illegal delegation with profit motive. This leads to the abuse of key-delegation in CP-HABE.

IV. PRELIMINARIES

A. ACCESS STRUCTURE

Here is a brief introduction to the AND-gate access structure with two values of positive and negative attributes.

Let \( N \) be a set of \( n \) attribute values, \( N = \{ t_1, t_2, \ldots, t_n \} \). For each attribute \( t_i \in N \), if it has a positive attribute, it is marked as \( +t_i \), and let the attribute value be 1; if it has a negative attribute, it is marked as \( -t_i \), and let the attribute value be 0. One AND-gate access structure is marked as \( W = \Lambda_{t_i \in T_i} \), where \( T \subseteq N \). By the assignment of the above processes 1 and 0, the value of the AND-gate \( W \) can be obtained. Given the attribute set \( T \), if the value of the AND-gate \( W \) equals to 1, the attribute set \( T \) satisfies the AND-gate access structure; otherwise, the attribute set \( T \) does not satisfy the AND-gate access structure.

B. BILINEAR GROUPS

Let \( G, G_T \) be cyclic groups of prime order \( p \), and let \( g \) be a generator of \( G \). We say \( G \) has an admissible bilinear map, \( e : G \times G \rightarrow G_T \), into \( G_T \) if the following three conditions hold:

- Bilinearity: \( e(g^a, g^b) = e(g, g)^{ab} \), for all \( a, b \in \mathbb{Z}_p \);
- Non-degeneracy: \( e(g, g) \neq 1 \);
- Computability: \( e(g^a, g^b) \) for all \( g^a, g^b \in G \) can be computed efficiently.

C. THE GENERIC BILINEAR GROUP MODEL

We prove the security of the proposed construction in the generic bilinear group model introduced by Shoup [33]. Let \( \xi_0 \) and \( \xi_1 \) be two random encodings of an additive group \( F_p \), they are injective maps: \( \xi_0, \xi_1 : F_p \rightarrow \{0, 1\}^m \), where \( m > 3 \log(p) \). We define \( G_i = \{ \xi_i(x) : x \in F_p \} \), for \( i = 0, 1 \). In the model, the adversary is given access to the oracles that simulate the hash function, group operations in \( G_0, G_1 \) and the non-degenerate bilinear map \( e : G_0 \times G_0 \rightarrow G_1 \). We let \( G_0 \) denote a generic bilinear group.

D. COMPLEXITY ASSUMPTIONS

\( l - SDH \) Assumption [34]: Let \( G \) be a bilinear group of prime order \( p \) and \( g \) be a generator of \( G \), the \( l \)-Strong Diffie-Hellman \( (l - SDH) \) problem in \( G \) is defined as follows: given a \( (l + 1) \)-tuple \( (g, g^x, g^{x^2}, \ldots, g^{x^l}) \) as inputs, output a pair \( (c, g^{1/(c+1)}) \in \mathbb{Z}_p \times G \). An algorithm \( A \) has advantage in solving \( l - SDH \) in \( G \) if

\[
\Pr\left[A\left(g, g^x, g^{x^2}, \ldots, g^{x^l}\right) = (c, g^{1/(c+1)})\right] \geq \varepsilon,
\]

where the probability is over the random choice of \( x \) in \( Z_p^\ast \) and the random bits consumed by \( A \).

Definition 1: The \( l - SDH \) assumption holds if the \( l - SDH \) challenge problem can not be solved by any polynomial-time algorithm \( A \) who has at least non-negligible advantage.

E. SYSTEM DESCRIPTION

In order to make our system more clearly, a framework of the proposed CP-HABE-AKDA scheme is given in Fig.3, which contains four entities:

- **Root Authority (RA):** It is responsible for generating the system public key \( PK \), master key \( MK \), and the level key \( SK_i \) for each domain authority. The root authority RA is the fundamental part of the entire architecture and is the ultimate trusted party.

- **Domain Authority (DA):** DA is responsible for generating the attribute private key \( SK_{i,\omega} \) for the user with the attribute set \( \omega \) at level \( i \), the level key \( SK_{i+1} \) for the...
next level domain authority by the key-delegation, and responsible for managing the users at level \( i \), including user additions and deletions.

- **User (U):** Users of each domain are managed by a directly affiliated domain authority. The user attribute private key in the proposed scheme is related not only to the user’s attribute set, but also to the user’s level. By embedding the relevant parameters of the two into \( SK_{i,\omega} \) at the same time, the user attribute private key held by the user not only realizes the user’s identity in the system, but also prevents the abuse of key-delegation.

- **Attribute:** When the attribute set satisfies the access structure, the domain authority at the right level generates the user attribute private key. The attribute set and user identity are not bound. Only an attribute set with hierarchy can represent a user.

In the proposed CP-HABE scheme, the root authority and domain authorities in each domain form a directed access graph, which is represented by \( V = (L_N, E) \). \( L \) is a set of vertices with a binary relationship. In the partially ordered set \( L_N = \{L_0, L_1, \ldots, L_n\} \), \( L_0 \) represents the root authority, \( L_1, L_2, \ldots, L_n \) stands for \( n \) domain authorities of different levels, \( E \) is a set of edges. In a hierarchy, the relationship of domain authorities can be modeled as a partial ordered set with a binary relationship. In the partially ordered set \( (L_N, \leq) \), if \( L_t \leq L_i \), it represents domain authority \( DA_i \) with a higher security permission than \( DA_t \). For example, domain authority \( DA_i \) can generate the level key of \( DA_t \), but not vice versa. If \( L_t \leq L_i \), then \( L_t \) is a successor of \( L_i \). Further, if \( L_t \leq L_i \), and there is no other \( L_k \) such that \( L_k \leq L_t \leq L_i \), then \( L_t \) is said to be an immediate predecessor of \( L_i \), and \( L_t \) is an immediate successor of \( L_i \).

**V. CP-HABE-AKDA**

In view of randomizing the delegation key parts, one of which are analyzed as causes of the key-delegation abuse in the CP-HABE scheme, this section proposes the ciphertext-policy hierarchical attribute-based encryption against key-delegation abuse (CP-HABE-AKDA) and introduces the related definitions, construction, and security analysis.

### A. CP-HABE-AKDA DEFINITION

This scheme consists of the following six algorithms:

- **GlobalSetup (\( \lambda, L \)):** The global initialization algorithm is run by the root authority \( RA \) with inputting the global security parameter \( \lambda \), generates the public key \( PK \) and the master key \( MK \) for a hierarchy of depth \( L \).

- **LevelKeyGeneration (\( PK, MK \)):** A level key generation algorithm is run by the root authority \( RA \) with inputting the global public key \( PK \) and the master key \( MK \), generates a level key \( SK_i \) for the domain authority \( DA_i \) at level \( i (1 \leq i \leq L) \).

- **LevelKeyDerivation (\( SK_i \)):** The algorithm is for the level key derivation. The domain authority \( DA_i \) at level \( i \) generates a level key \( SK_h \) for the domain authority at level \( h \) according to the key of level \( i \) and the public information between the levels. Authorities of each level domain use the level key derivation algorithm for key-delegation, which provides scalability for the solution.

- **AttributeKeyGeneration (\( SK_i, PK, \omega \)):** The algorithm is to generate the attribute private key for users at each level. The domain authority generates an attribute private key \( SK_{i,\omega} \) according to the public key \( PK \), the level key \( SK_i \) and the attribute set \( \omega \) at level \( i \), and is used to decrypt the ciphertext. The level key is used to generate the attribute private key of the level, which ensures the correctness when decrypting.

- **Encryption (\( PK, M, \tau, i \)):** The encryption algorithm takes message \( M \), access structure \( \tau \) and public key \( PK \) at the level \( i \) as inputs, and the output is ciphertext \( CT \).

- **Decryption (\( CT, SK_{i,\omega} \)):** The decryption algorithm takes the attribute private key \( SK_{i,\omega} \) and ciphertext \( CT \) as the input. If the attribute set owned by the user meets the access structure and the level requirements of ciphertext, the plaintext will be output.

### B. IND-S-CPA SECURITY MODEL

The general security model defined below describes the interaction between the adversary and the encryption algorithm. The formal security model between the adversary and the challenger is defined as follows:

- **Init:** The challenger runs the initialization algorithm and sends the public parameters obtained by the algorithm to the adversary, while retaining the master key.

- **Phase 1:** The adversary is allowed to execute the following types of polynomial bounded number of key queries:

  - **Type 1:** The adversary queries the domain authority of level \( i \) for the key with attribute set \( \omega_1, \omega_2, \ldots, \omega_q \). The challenger sends the attribute private key \( SK_{i,\omega_r} \) to the adversary.

  - **Type 2:** The adversary queries the domain authority of level \( i \) for the key with attribute set \( \omega_1, \omega_2, \ldots, \omega_q \). The challenger sends the attribute private key \( SK_{i,\omega_r} \) to the adversary.

- **Challenge:** The adversary submits two equal-length plaintext messages \( M_0 \) and \( M_1 \) to the challenger as the ciphertext that they want to be challenged. In addition, the adversary also gives the challenger an access policy \( \Gamma \). None of the queried keys generated in phase 1 can satisfy the access policy \( \Gamma \). The challenger tosses a coin \( b \in \{0, 1\} \) randomly, then encrypts \( M_b \) with the access policy \( \Gamma \), and returns ciphertext \( CT^* \) to the adversary.

- **Phase 2:** Repeat phase 1. The adversary continues to submit key queries to the challenger, and the queried keys do not comply with the access policy \( \Gamma \). The challenger returns the corresponding keys to the adversary.

- **Guess:** The adversary outputs a guess value \( b' \) for \( b \). if \( b' = b \), the adversary will win the game. The advantage of the adversary’s success in this game is defined as \( \Pr \{b' = b\} - \frac{1}{2} \).
Definition 2: In the CP-HABE-AKDA security game, if all polynomial-time adversaries have at most a negligible advantage, the CP-HABE-AKDA is IND-s-CPA secure.

C. SECURITY MODEL AGAINST KEY-DELEGATION ABUSE ATTACK

The security definition against key-delegation abuse in a CP-HABE-AKDA scheme is described by a security game between a challenger and an adversary, as follows:

- **Init:** The challenger runs the initialization algorithm and sends the public parameters obtained by the algorithm to the adversary, while retaining the master key. The total attribute set \( \mu \) and the message space \( M \) are also defined in this phase.

- **Query 1:** The adversary queries the challenger for the domain authority key \( SK_i \) at level \( i \). When \( 1 \leq i \leq L \), the challenger runs LevelKeyGeneration, generates a level key \( SK_i \) for the domain authority \( DA_i \), and then sends it to the adversary.

- **Query 2:** The adversary uses the level key \( SK_i \) obtained in the Query 1 to perform a level key query at level \( h \) \((1 \leq h \leq L)\) domain authority \( DA_h \). The adversary sends \( SK_i \) to the challenger, queries the level key \( SK_h \) at level \( h \). The challenger runs the algorithm LevelKeyDerivation to get the level key \( SK_h \) at level \( h \), and then sends it to the adversary.

- **Query 3:** The adversary queries the challenger for the attribute private key of the different attribute set \( \sigma_1, \ldots, \sigma_Q \subseteq \mu \) at level \( h \). When \( 1 \leq \sigma \leq Q \), for each \( \sigma \), the challenger runs the algorithm AttributKeyGeneration and generates the attribute private key \( SK_h, \sigma \), and sends it to the adversary. The adversary can query the challenger adaptively.

- **Output:** The adversary selects a new attribute set \( \omega^* \)

  - for level \( h \), while \( \omega^* \neq \sigma \), \( 1 \leq \sigma \leq Q \). And the adversary generates a new private key \( SK_h, \omega^* \) and a new general decryption algorithm \( Decryption^*(CT, SK) \) for attribute set \( \omega^* \) at level \( h \) \((1 \leq i \leq h \leq L)\), then sends them to the challenger.

The adversary will win the game if the following conditions are met:

1. \( Decryption^*(SK_h, \omega^*, Encryption(PK, M, \tau, h)) = M \)

2. \( Decryption'(SK_h, \omega^*, Encryption(PK, M, \tau, h)) = \bot \)

The advantage of an adversary is defined to be the probability that the adversary wins the security game.

Definition 3: In this security game, if all polynomial-time adversaries have at most one negligible advantage, the CP-HABE-AKDA scheme is secure against key-delegation abuse attack.

D. CP-HABE-AKDA CONSTRUCTION

In this section, we shall present our CP-HABE-AKDA scheme, the workflow of which is shown in Fig.4. The proposed scheme introduces a directed graph with the hierarchical structure to achieve delegation between upper and lower hierarchical domain authorities. When constructing an attribute private key, the algorithm associates the key parts related to the attribute set with the parameters related to the level, so that the domain authorities could not use key-delegation and parts of attribute private key for forging a new key, which would result in key-delegation abuse.

- **GlobalSetup:** Given a security parameter \( \lambda \) and an attribute universe \( \mu \) with the size of \( n \), where \( n \) is a natural number. The entire attribute set is \( \mu := \{1, 2, \ldots, n\} \). The algorithm first chooses two bilinear groups \( G \) and \( G_T \) of prime order \( p \), two random generators \( g_0, h_0 \in G \), a bilinear map \( e : G \times G \rightarrow G_T \). Let \( H : Z_p \times G \rightarrow \{\{0, 1\}\}^d \rightarrow G \) be a secure hash function. The algorithm randomly selects \( \alpha, \beta \in Z_p \), and computes \( Y = e(g_0, h_0)^{\alpha - \beta}, g_1 = g_0^\alpha, g_2 = g_0^\beta \). In addition, it selects random elements \( y_1, y_2, \ldots, y_\Psi \in Z_p, 1 \leq \Psi \leq L \) and \( g_3, f_1, f_2, \ldots, f_L \in G \). Root authority RA broadcast public key \( PK \), keep the master key \( MK \):

\[
PK = (p, G, G_T, e, g_0, h_0, g_3, f_1, f_2, \ldots, f_L, H, Y, \mu)
\]

\[
MK = (\alpha, \beta, g_1, g_2, B = \{y_1, y_2, \ldots, y_\Psi\})
\]

- **LevelKeyGeneration:** Root authority RA assigns a unique public identity \( LID_i \in B \) and a secret random large positive integer \( y_\phi \in Z_p \) for each level. Then the key parameter at level \( i \) is \( k_i = H(y_\phi, LID_i) \). RA stores \( k_i \) and \( LID_i \) in the entry of \( i^{th} \) level key table, selects a random number \( r \in Z_p \), and generates the level key of
domain authority $DA_i$ as follows:

$$SK_i = \left( g_0^{b-k_i+r}, h_0^{k_i}, g_0^{a+r} \cdot \left( g_3 \prod_{l=1}^i f_l \right)^{k_i} \right)$$

The level key $SK_i$ and key parameter $k_i$ are sent to $DA_i$ through a secure channel between $RA$ and $DA_i$.

- **LevelKeyDerivation:** It is assumed that the level $i$ is the predecessor of the level $h$, $1 \leq i \leq h \leq L$, and their key parameters are $k_i$ and $k_h$ respectively. $DA_i$ uses the public information preset by $RA$ to generate a new level key, and achieves level key-delegation.

1. The root authority $RA$ randomly selects a large positive integer $q$ to publicly store derived information $y_{i,h} = k_h - H(k_i, LID_h) \mod q$ for the relationship $(V_i, V_h)$, which is from level $i$ to level $h$. And $q$ is published by $RA$.

2. According to the public information $LID_h$ and $y_{i,h}$, $DA_i$ at level $i$ uses its key parameter $k_i$ to calculate the value of key parameter $k_h$ at level $h$, that is $k_h = y_{i,h} + H(k_i, LID_h) \mod q$.

3. The domain authority $DA_i$ selects the random number $r' \in Z_p$, and uses its level key $SK_i$ to generate the level key $SK_h$ related to level $h$, as follows:

$$SK_h = \left( g_0^{b-k_h+r+r'}, h_0^{k_h}, g_0^{a+r+r'} \cdot \left( g_3 \prod_{l=1}^i f_l \right)^{k_h} \right)$$

- **AttributeKeyGeneration:** This algorithm is run by domain authority $DA_i$ at level $i$ to generate the attribute private key $SK_{i,\omega}$ for the users with an attribute set $\omega \subseteq \mu$. $DA_i$ selects secret random number $t_1, t_2, \ldots, t_{2n} \in Z_p$, and computes the public parameters $h_1 = h_0^{t_1}, \ldots, h_n = h_0^{t_{2n}}$, $h_{n+1} = h_0^{t_{2n+1}}, \ldots, h_{2n} = h_0^{t_{2n+2}}$.

In addition, $n - 1$ random values $x_1, x_2, \ldots, x_{n-1}$ are randomly chosen in $Z_p$, and associated with the level key parameter $k_i$ by computing $x_n = k_i - x_1 - \cdots - x_{n-1} \in Z_p$.

For each attribute $j \in \omega$, a random number $x_n$ is assigned as follows:

$$D_{i,\omega} = \begin{cases} g_0^{t_j} & (j \in \omega) \\ g_0^{x_n-t_j} & (j \in \mu \setminus \omega) \end{cases}$$

The attribute private key $SK_{i,\omega}$ for the attribute set $\omega$ at level $i$ is:

$$SK_{i,\omega} = \left( g_0^{b-k_i+r}, h_0^{k_i}, h_0^{a+r} \cdot \left( g_3 \prod_{l=1}^i f_l \right)^{k_i} \left( D_{i,\omega} \right)_{j \in \mu}, D' = g_0^{k_i} \right)$$

- **Encryption:** This algorithm encrypts a message $M \in G_T$ under the access control policy specified with an access structure $\tau = \Lambda_{j \in W} j$ at level $i$, where $W \subseteq \omega$. The encryption algorithm first selects a random value $s \in Z_p$. Then, the ciphertext $CT$ is composed as follows:

$$CT = \left( \left( E' = MY^{(a-b)s}, h_0^s, \prod_{j=1}^{\omega} g_0 \right), E_i, \omega, h_0^{s+j}, E_{i,\omega} = h_0^{k_j} \right)_{j \in W \setminus \omega}$$

- **Decryption:** There are two requirements on successful decryption: first, the attribute set associated with the user needs to satisfy the access structure; second, the attribute set is at the correct level. Therefore, the decryption algorithm consists of two steps: the first step confirms whether the attribute set $\omega$ satisfies the access structure $\tau$; and the second step uses the attribute private key of the correct level for message recovery. The decryption algorithm uses algorithm $DecAtt (CT, SK_{i,\omega})$ to perform the following steps:

1. For all attributes $j \in \mu$, if the set $\omega \subseteq \mu$ composed of $j$ conforms to the access structure $\tau$, the algorithm $DecAtt (CT, SK_{i,\omega})$ outputs $e (g_0, h_0)^{X_{\omega}}$. If such a set does not exist and the access structure is not satisfied, the algorithm outputs $\bot$.

$$DecAtt (CT, SK_{i,\omega}) = \prod_{j \in W \cup \{ \mu \setminus \omega \}} e (D_{i,\omega}, E_{i,\omega}) \prod_{j \in \omega} e (D_{i,\omega}, E_{i,\omega})$$

$$= \prod_{j \in \omega} e \left( g_0^{t_j}, h_0^{t_j} \right) \prod_{j \in \mu \setminus \omega} e \left( g_0^{x_n-t_j}, h_0^{t_j} \right)$$

$$= \prod_{j \in \omega} e \left( g_0^{t_j}, h_0^{t_j} \right) \prod_{j \in \mu \setminus \omega} e \left( g_0^{x_n-t_j}, h_0^{t_j} \right)$$

$$= e (g_0, h_0)^{s \sum_{j \in \omega} t_j} = e (g_0, h_0)^{s} = Z(1)$$

2. After confirming that the attribute set satisfies the access structure, the decryption algorithm executes:

$$Z(2) = e \left( g_0^{b-k_i+r}, h_0^s \right) = e (g_0, h_0)^{(b-k_i+r)s}$$

$$Z(3) = e \left( h_0^{a+r} \cdot \prod_{l=1}^{\omega} f_l \right)^{k_i} \prod_{l=1}^{\omega} e \left( h_0^{a+r}, f_l \right)^{k_i}$$

$$= e \left( h_0^{a+r} \cdot \prod_{l=1}^{\omega} f_l \right)^{k_i} = e \left( h_0, g_3 \cdot \prod_{l=1}^{\omega} f_l \right)^{k_i}$$

The correct $Z(3)$ value can only be recovered by users at the correct level in the hierarchy. Finally, if the user’s attribute private key satisfies access structure $\tau$ and has the correct level, the decryption algorithm
VI. SECURITY ANALYSIS OF CP-HABE-AKDA

The generic bilinear group model and the random oracle model are used to demonstrate that there is no effective adversary can break the security of the proposed scheme with non-negligible probability, if an adversary performs generic operations on the group used in the proposed scheme. In a non-negligible probability, if an adversary performs generic group model, group elements are encoded as a unique random string so that adversaries can manipulate group elements using canonical group operations in $G$ and $G_T$, and cannot test any property other than equality.

First, the functions and symbols used in the simulation security game are given. Define two kinds of random codes $\xi_0, \xi_1$ on additive group $F_p$, such that $\xi_0, \xi_1 : F_p \to \{0, 1\}^\mu$, as injective maps, where $\mu > 3 \cdot \log(p)$. Let $\nu = 0, 1, G_\nu = \{\xi_\nu(\rho) : \rho \in Z_p\}$. $G$ is a generic bilinear group, $\xi_0(\rho)$ represents $g^\rho \in G$, $\xi_1(\rho)$ represents $e(g, g)^\rho \in G_T$.

The challenger $\Xi$ receives two oracle models to compute group operations on $G$ and $G_T$, one oracle model to compute non-degenerate bilinear mappings $e : G \times G \to G_T$, and the other oracle model represents hash function $H : \{0, 1\}^s \to G$.

A. SECURITY PROOF OF CONSTRUCTION

Theorem 1: Let $q$ be an upper bound on the total number of group elements that an adversary $A$ can receive from queries she makes to the challenger $\Xi$ for elements from the oracle model for the hash function, groups $G$ and $G_T$, bilinear map $e$, and from his interaction in the CP-HABE-AKDA security game. The advantage of the adversary in the security game is $O(q^2/p)$.

Proof: Here we first bind the advantage of adversary $A$ to an improved security game. In the security game defined in Definition 2, the ciphertext that challenger $\Xi$ randomly outputs contains the message-related element $\hat{C}$, which might be $\hat{C} = M_0 \cdot e(g_0, h_0)^{(a-\beta)s}$ or $\hat{C} = M_1 \cdot e(g_0, h_0)^{(a-\beta)s}$. In an improved security game, the challenger $\Xi$ may return $\hat{C}$ as $e(g_0, h_0)^{(a-\beta)s}$, or $\hat{C} = e(g_0, h_0)$, where $\theta$ is uniformly selected from $Z_p$, and the adversary $A$ has to determine which value to take. Obviously, any adversary with an advantage $\epsilon$ in the CP-HABE-AKDA security game can be converted to an adversary with at least an advantage $\epsilon/2$ in an improved security game. Therefore, the advantage of the adversary can be bound in the improved security game. The following will prove that in an improved security game, no adversary $A$ has a non-negligible advantage. Here we define all possible queries and corresponding outputs that adversary $A$ might perform in a secure game simulation. In the security game, adversary $A$ receives the following encodings from the interaction with the challenger $\Xi$.

In the initialization phase of the security game, the components generated by algorithm $\text{GlobalSetup}$ include:

1. $\xi_0(\nu) \mapsto g_0 = g^\nu, \xi_0(\nu) \mapsto h_0 = g^\nu$;
2. $\xi_0(\nu \alpha) \mapsto g_1 = g_0^\nu, \xi_0(\nu \beta) \mapsto g_2 = g_0^\beta = g^{\nu \beta}$;
3. $\xi_0(\nu \eta) \mapsto f_1 = g_0^{\nu}, \xi_0(\nu t) \mapsto H(j) = \xi_1(\nu \alpha) \cdot A = e(g_0, h_0)^{(a-\beta)}$.

The challenger sends the public parameters to $A$.

In Phase 1 and Phase 2 of the security game, the components generated by the attribute private key algorithm $\text{AttributeKeyGeneration}$ associated with an attribute set $\omega$ at level $i$ include:

1. $\xi_0(\nu t_j) \mid j \in \mu \mapsto h_j = h_0^{\nu t_j} = g^{\nu t_j}$;
2. $\xi_0(\nu t_{n+j}) \mid j \in \mu \mapsto h_j = h_0^{\nu t_{n+j}} = g^{\nu t_{n+j}}$;
3. $\xi_0(\nu (\beta - k_1 + r)) \mapsto g_0^{\beta - k_1 + r} = g^{\nu (\beta - k_1 + r)}$;
4. $\xi_0(\nu (\alpha + r)) \mapsto g_0^{\alpha + r} \cdot (g_3 \cdot \sum_{i=1}^{\ell_1} f_i) = g^{\nu (\alpha + r) + \sum_{i=1}^{\ell_1} f_i}$;
5. $\xi_0(\nu (\theta_{t j}) \mid j \in \omega) \mapsto D_j = g_0^{\nu \theta_{t j}} = g_0^{\nu \theta_{t j}}$;
6. $\xi_0(\nu (\theta_{t j}) \mid j \in \mu \setminus \omega) \mapsto D_j = g_0^{\nu \theta_{t j}} = g_0^{\nu \theta_{t j}}$.

In the challenge phase of the security game, the components generated by algorithm $\text{Encryption}$ include:

1. $\xi_0(\nu \theta) \mapsto \hat{C} = e(g_0, h_0)^\theta, \xi_1((\alpha - \beta) \nu s) \mapsto \hat{C} = e(g_0, h_0)^{(a-\beta)s}$;
2. $\xi_0(\nu s) \mapsto h_0^{\nu s} = g^{\nu s}, \xi_0(\nu s \theta) \cdot \xi_0(\nu s \sum_{i=1}^{\ell_1} \eta_i) \mapsto g_3 \cdot \sum_{i=1}^{\ell_1} f_i = g^{\nu s + \sum_{i=1}^{\ell_1} f_i}$;
3. $\xi_0(\nu s t_j) \mid j \in W \mapsto E_{s \omega} \hat{C} = h_j^{\nu s} \mid j \in W = g^{\nu s t_j} \mid j \in W$;
4. $\xi_0(\nu s t_j) \mid j \in \mu \setminus W \mapsto E_{\mu \setminus \omega} \hat{C} = h_j^{\nu s} \mid j \in \mu \setminus W = g^{\nu s t_j} \mid j \in \mu \setminus W$.

It should be noted that if $A$ queries for a secret key satisfying the challenge access policy $\Gamma^*$ then the challenger $\Xi$ would not issue the key. The adversary can use the group elements received from the interaction with the challenger to perform generic group operations and equality tests on $\{\alpha, \beta, \gamma, \nu, \lambda, \theta, r, s, x_1, \ldots, x_{n-1}, l_j, \eta_j\}$, where each variable is an element from $Z_p$ picked randomly in our scheme. Queries that the adversary can execute include:

- Queries to oracle model for group operations on $G$ and $G_T$. When adversary $A$ requests for multiplication or division of group elements represented by their random encodings, the oracle model returns the addition or subtraction in the additive or multiplicative cyclic groups respectively, based on the encoding of the group elements.
- Queries to the oracle for computing pairing operation. When the adversary $A$ requests for pairing of the group elements represented by random encodings, the oracle
model returns the multiplication of group elements in the multiplicative cyclic groups.

Next, it will be proved that by using the above simulation results, adversary $A$ cannot distinguish the challenger’s ciphertext between $C = e(g_0, h_0)^\theta$ and $\hat{C} = e(g_0, h_0)^{(\alpha - \beta)s}$ with a non-negligible advantage.

According to the standard approach of the security in the generic group model, when the adversary queries for the group oracle model, an unexpected collision occurs because the challenger randomly selects values for the variables $\{x, y, y_0, y'_0, t, r, s, x_1, \ldots, x_{n-1}, t_1, t_2\}$. That means, when two different forms of rational function correlation queries are conducted, the two query values will be the same due to random selection of variable values. In the security game of the generic group model, the probability of a collision with any two different queries is the same as the adversary’s view, which is at most $O(q^2/p)$. Therefore, for the output of ciphertext $\hat{C}$, when $\theta = (\alpha - \beta)s$, the adversary can complete the security game with a non-negligible advantage.

The following will use reduction to absurdity for constructing complete the security game with a non-negligible advantage.

According to the standard approach of the security in the group oracle model, an unexpected collision occurs because the challenger randomly selects values for the variables $\{x, y, y_0, y'_0, t, r, s, x_1, \ldots, x_{n-1}, t_1, t_2\}$. That means, when two different forms of rational function correlation queries are conducted, the two query values will be the same due to random selection of variable values. In the security game of the generic group model, the probability of a collision with any two different queries is the same as the adversary’s view, which is at most $O(q^2/p)$. Therefore, for the output of ciphertext $\hat{C}$, when $\theta = (\alpha - \beta)s$, the adversary can complete the security game with a non-negligible advantage.

The following will use reduction to absurdity for constructing complete the security game with a non-negligible advantage.

The polynomial in (1) also contains terms $B$, $C$, $E$, hence it needs to cancel terms $B$, $C$ to construct a polynomial of the form $(\alpha - \beta)s$, and also require term $E$ to be 0. The following is a case-based analysis of the strategy that may be deployed by adversary $A$ to cancel terms $B$, $C$ and make term $E$ be 0. The analysis is based on the fact that adversary $A$ can never use a key that satisfies access control policy $\Gamma^*$. In order to satisfy $\Gamma^*$, the key should correspond to the correct level $i$ and satisfy the access structure $\tau^*$. Then we identify the following two situations:

- **Case 1** In this case, the adversary $A$ uses the output obtained from the type 2 query in the security game defined in this section. The key obtained from the query corresponds to the level $i$ in the hierarchy but does not satisfy $\tau^*$.

  In order to cancel terms $B$, $C$ of (1), the adversary $A$ can pair the query results of the units $(5, 1)$ and $(8, 2)$ in Table 2 and construct a polynomial of the form $-t - \sum_{l=1}^{i} \eta_l$. In order to make term $E$ to be 0, then

\[ k_i = 0, \quad \text{DecAtt} \left( CT, SK_i,\omega \right) = e(g_0, h_0)^{k_i} = Z^{(1)} \]

is a fixed value. This means that when term $E$ is 0, the adversary $A$ can make a key that satisfies $\tau^*$, which contradicts the assumption of case 1. According to the security definition, adversary $A$ should not have at least one element of key associated with the ciphertext created by using $\tau^*$. Therefore, term $E$ cannot be 0, so the adversary $A$ cannot use the strategy of case 1, for constructing a polynomial of the form $(\alpha - \beta)s$.

- **Case 2** In this case, the adversary $A$ uses the output obtained from the type 2 query of the security game in this section. The key obtained by the adversary from the query does not correspond to level $i$ in the hierarchy (i.e., the user does not belong to the correct level), but instead satisfies the access structure $\tau^*$ (i.e., the user has the required attributes).

Suppose that adversary $A$ can only use the key corresponding to level $\hat{i}$ in the hierarchy, where $\hat{i} \neq i$, and create a challenge ciphertext for it. However, these keys satisfy the partial access strategy $\Gamma^*$, that is, only the access structure $\tau^*$ is satisfied. This assumption is consistent with the security definition that adversary $A$ cannot access the key that satisfies the challenge access policy $\Gamma^*$. In order to distinguish the keys corresponding to levels $\hat{i}$ and $i$, the level key parameters $k_\hat{i}$ and $k_i$ are used here, respectively.

Since the related secret component which satisfies the AND access structure $\tau^*$ can be accessed, in this case, the adversary $A$ can cancel term $E$. To cancel terms $B$ and $C$, adversary $A$ can pair the queries in Table 2 with cells $(4, 1), (5, 1)$ and $(8, 3)$, and construct a polynomial of the form $-k_{\hat{i}}t - k_i \sum_{l=1}^{i} \eta_l$. To cancel terms $B$ and $C$, adversary $A$ needs to query $k_i$. However, this contradicts the security definition. Therefore, the adversary cannot construct a polynomial of the form $(\alpha - \beta)s$, by using the strategy of Case 2.
In summary, it is concluded that the adversary A cannot perform a polynomial query to obtain a polynomial of the form \((\alpha - \beta) s\). This proves that there is no generic adversary A could break the security of the proposed CP-HABE-AKDA with a non-negligible advantage.

**B. PROOF OF CONSTRUCTION AGAINST KEY-DELEGATION ABUSE**

**Theorem 2:** The proposed CP-HABE-AKDA scheme is secure against Key-delegation Abuse Attack with respect to Definition 3 in the generic group model.

The proof is also based on the generic group model. Suppose the adversary generates a new key \(SK_h,\omega^*\) for the attribute set \(\omega^*\) at level \(h\), through the level key derivation algorithm and the attribute private key generation algorithm, and gives the decryption algorithm \(Dec_{\omega^*}(\cdot)\), which can decrypt the ciphertext encrypted with access policy \(\Gamma^*\). First, we use the standard parameters of the general group model, it is to be proved that if this occurs with a non-negligible probability, then the multilinear polynomials in the new key as described above, must also meet the corresponding constraints. Therefore, we use reduction to absurdity to assume that the multilinear polynomial corresponding to the adversary’s output satisfies the required constraints, and then we get a contradiction. In order to satisfy the constraints, polynomials must have a specific structure, namely, they can only depend on certain given group elements.

**Proof:** For a generic-group adversary, the challenger executes a security game against key-delegation abuse attack as follows. For each group element related to the adversary, the challenger tracks its discrete logarithm through a multi-variate rational function in an indeterminate formal variable. The simulation process associates each group element with some rational function.

For each distinct rational function in the set, it enters the corresponding encoding \(\omega_0\) or \(\xi_1\) and provides the result to the adversary as the encodings of the particular group elements. In this security game, the adversary, the codes received by the adversary from the interaction with the challenger, are based on the generic group model, as the same as in the proof of theorem 1, so they will not be repeated here.

During the output phase of the security game, the adversary outputs a claimed new attribute private key \(SK_{h,\omega^*}\) and a suitable decryption algorithm \(Dec_{\omega^*}(\cdot)\) for the new attribute set \(\omega^*\) at level \(h\). For ciphertext \(CT\) encrypted with access structure \(\tau = \land_{j \in W_j}\) at level \(h\), if the new private key can be used to compute \(Y = e(g_0^*, h_0^*)(alpha - beta)^s\), thus it can decrypt \(M\) from \(E'\). Therefore, the new private key \(SK_{h,\omega^*}\) should contain the group element of each attribute \(j\) to be paired with the group element corresponding to \(E_i,\omega\) (or \(E_i',\omega\)) of the ciphertext in the bilinear map. For the attribute \(j\) in \(\mu\), these group elements are represented by \(D_f^j\), and the necessary structure of the new key \(SK_{h,\omega^*}\) can be expressed as \(\beta')_j \in \mu\). After the necessary structure of the new key satisfies the winning condition, the adversary can construct another new key differently, which means that the adversary can construct a new key component \(D^*_h,\omega\) using a linear combination of the encoded rational functions.

Let \(\mu = \sum_{j=0}^{n-1} \sum_{\omega = 0}^{\omega} \sum_{\omega = 0}^{\omega^*} \delta_j h_0^j \sum_{\omega = 0}^{\omega^*} \delta_n^j + \sum_{\omega = 0}^{\omega^*} \delta_{n+j} h_n^j + \sum_{\omega = 0}^{\omega^*} \delta_{n+j} h_{n+j} \) be denoted as \(D^*_h,\omega\). In order to win the game, \(D^*_h,\omega\) must meet the following conditions:

1. \(\sum_{j \in \mu} \sum_{\omega \in \omega} \delta_j,\omega x_j^0 = k_h^*\),
2. When \(1 \leq i < h \leq L\), \(\forall j \in \omega^*\), \(\sum_{\omega \in \omega} \delta_j,\omega x_j^0 = 0\);
3. When \(1 \leq i < h \leq L\), \(\forall j \in \omega^*\), \(\sum_{\omega \in \omega} \delta_j,\omega x_j^0 = 0\) and \(\sum_{j \in \omega} \delta_j,\omega x_j^0 \neq 0\).

Assuming the new key meets the above conditions, perform the following steps:

1. Considering condition 1, \(\sum_{j \in \mu} \sum_{\omega \in \omega} \delta_j,\omega x_j^0 = k_h^*\), since \(\sum_{j \in \mu} \sum_{\omega \in \omega} \delta_j,\omega x_j^0 = k_h^*\) uniformly selected randomly in \(Z_p\), there are coefficients \(\delta_1,\omega = \delta_2,\omega = \cdots = \delta_n,\omega\), which can be represented by \(\delta_0\).
2. Considering condition 2, when \(1 \leq i < h \leq L\), for all conditions \(\forall j \in \omega^*\), \(\sum_{\omega \in \omega} \delta_j,\omega x_j^0 = \sum_{\omega \in \omega} \delta_j,\omega x_j^0 = 0\), since \(x_j^0\) uniformly selected randomly in \(Z_p\), the following conclusion can be drawn: if \(j \in \omega^*\) and \(j \notin \omega\), then \(\delta_0 = 0\); then it can be derived, when \(1 \leq i < h \leq L\), if \(\delta_0 \neq 0\), then \(\omega^* \subseteq \omega_0\).
3. Considering condition 3, when \(1 \leq h < i \leq L\), for all conditions \(\forall j \notin \omega^*\), \(\sum_{\omega \in \omega} \delta_j,\omega x_j^0 = \sum_{\omega \in \omega} \delta_j,\omega x_j^0 = 0\) and \(\sum_{j \in \omega} \delta_j,\omega x_j^0 = \sum_{j \in \omega} \delta_j,\omega x_j^0 = 0\), since \(x_j^0\) is uniformly randomly selected in \(Z_p\), the following conclusions can be drawn: if \(j \in \omega^*\) and \(j \notin \omega\), \(\delta_0 = 0\), then it can be derived, when \(1 \leq h < i \leq L\), if \(\delta_0 \neq 0\), then \(\omega^* \subseteq \omega_0\).

Therefore, when \(h = i\), if \(\delta_0 \neq 0\), then \(\omega^* \subseteq \omega_0\). This will cause that the adversary fails to generate a new key for \(\omega^* \neq \omega_0\) at level \(h (1 \leq i < h \leq L)\), or the ciphertext decrypted with the key of the attribute set \(\omega_0\) can also be decrypted with the generated new key. Thus, it contradicts the conditions in the security definition. Therefore, the assumption is not true. The adversary cannot successfully generate a new private key to win the game. This proves that the proposed scheme can resist the key-delegation abuse attack.

**VII. TRACEABLE CP-HABE-AKDA**

A secure IoT-connected healthcare system based on CP-HABE is expected to implement key delegation in daily situations, helping doctors save time in treating patients; in case of key leakage, it would be able to track the identity of the key leaker and protect patients’ privacy. We present a CP-HABE-AKDA scheme with white-box traceability for the IoT-connected healthcare system. To better illustrate our scheme, the system model, scheme definition, security model, traceability model and scheme construction are described respectively in this section.

As shown in Fig.5, our IoT-connected healthcare system mainly involves five various types of entities: root authority(e. g., hospital administrative department), domain
hospital management department to the pharmacy department an attribute private key for pharmacy Doctor B (step 6). A delegates the Pharmacy Department (step 5) to generate to the prescription. Then the outpatient department of Doctor then he uploads the physiological data and preliminary diagnosis and treatment on the patient in the ambulance, recover the encrypted data (step 7).

When a drug supplier purchases prescriptions from a doctor at a high price, the doctor may give the attribute private key to the drug supplier for the benefit. If a key’s leaking is discovered, the hospital management department will trace the identity of the doctor based on the leaked key (step 8).

A. TRACEABLE CP-HABE-AKDA DEFINITION

The traceable CP-HABE-AKDA scheme consists of the following eight algorithms.

- GlobalSetup ($\lambda, L$): The global initialization algorithm is run by the root authority RA with inputting the global security parameter $\lambda$, generates the global public key $PK$ and the master key $MK$ for a hierarchy of depth $L$.
- USignKeyGen ($PK$): The user sign-verify key generation algorithm, takes $PK$ as input, and outputs secret sign-key $x_U$ and public verify-key $P_U$.
- LevelKeyGeneration ($PK, MK$): A level key generation algorithm is run by the root authority RA with inputting the global public key PK and the master key MK, generates a level key $SK_i$ for the domain authority $DA_i$ at level $i (1 \leq i \leq L)$.
- LevelKeyDerivation ($SK_i$): The level key derivation algorithm is run by the domain authority $DA_i$ of level $i$, which generates a level key $SK_i$ for the domain authority of level $h (1 \leq i < h \leq L)$ according to the level key $SK_i$ and the public information between the levels. Authorities of each level domain use the level key derivation algorithm for key-delegation.
- AttributeKeyGeneration ($SK_i, PK, \omega$): The algorithm is to generate the attribute private key for users at each level. The domain authority generates an attribute private key $SK_{i,\omega}$ according to the public key $PK$, the level key $SK_i$ and the attribute set $\omega$ at level $i$, and is used to decrypt the ciphertext. The level key is used to generate the attribute private key of the level, which ensures the correctness when decrypting.
- Encryption ($PK, M, \tau, i$): The encryption algorithm takes message $M$, access structure $\tau$ and public key $PK$ at the level $i$ as inputs, and the output is ciphertext $CT$.
- Decryption ($CT, SK_{U,\omega}$): The decryption algorithm takes the attribute private key $SK_{U,\omega}$ and ciphertext $CT$ as the inputs, and outputs $M$ or $\bot$.
- Trace ($PK, SK_{U,\omega}$): Identity tracking algorithm takes the user attribute private key $SK_{U,\omega}$ and $PK$ as inputs, RA checks the key elements, and outputs a valid flag 1 or invalid flag 0. If the output is 1, RA queries the identity information.

B. IND-S-CPA SECURITY MODEL

In this paper, the security game between a challenger and an adversary is used to describe the ciphertext indistinguishability under adaptive chosen plaintext attacks (IND-s-CPA) of
a traceable CP-HABE-AKDA scheme. In this model, RA is not allowed to conspire with DA. The adversary chooses to use the access policy $\Gamma^*$ for the encryption challenges. The access policy $\Gamma^*$ includes the partial level $i$ of a certain level range and the AND-gate access structure where the attribute set $\omega$ belongs. The formal security model is defined between the adversary and the challenger as follows:

- **Init**: The adversary submits a challenge access policy $\Gamma^*$ to the challenger.
- **Setup**: The challenger runs algorithms $\text{GlobalSetup}$ and $\text{USignKeyGen}$, then sends the public key $PK$ obtained by the algorithm to the adversary.
- **Phase1**: The adversary queries the challenger for a secret sign-key corresponding to a series of honest identities $U$. Since user identity $U$ is bound to attributes, here the user is corresponding to the attribute set $\omega$ at level $i$. If identity $U$ is colluded, the adversary submits the public verify-key directly to the challenger. The adversary queries the challenger for the decryption key corresponding to the identity $U$. The adversary is allowed to execute the following types of polynomial bounded number of key queries:
  - Type 1: The adversary queries the domain authority of level $i$ for the key with attribute set $\omega_1, \omega_2, \ldots, \omega_q$. The challenger sends the attribute private key $SK_{i,\omega_r}, \forall r \in \{1, 2, \ldots, q\}$ to the adversary.
  - Type 2: The adversary queries the domain authority of level $i$ ($i \neq i$) for the key with attribute set $\omega_1, \omega_2, \ldots, \omega_q$. The challenger sends the attribute private key $SK_{i,\omega_r}, \forall r \in \{1, 2, \ldots, q\}$ to the adversary.
- **Challenge**: The adversary outputs a guess value $b'$ for $b$. If $b' = b$, the adversary will win the game. The advantage of the adversary’s success in this game is defined as $\Pr[b' = b] - \frac{1}{2}$.

Definition 4: In the traceable CP-HABE-AKDA security game, the traceable CP-HABE-AKDA is IND-s-CPA secure, if all polynomial-time adversaries have at most a negligible advantage.

### C. TRACEABILITY MODEL

The user identity information in the scheme proposed in this section is related to the user attribute private key through the user’s signature. Therefore, a white-box traceability security game is similar to the unforgeability of a signature. It can be described by the following security game between challenger and adversary.

- **Init**: The challenger runs $\text{GlobalSetup}$ and $\text{USignKeyGen}$ algorithms and sends the public key obtained by the algorithm to the adversary.
- **Phase1**: The adversary is allowed to perform a polynomial bounded number of key queries: the adversary queries the challenger for the signature key related to a series of honest identity $U$, that is, the adversary submits the attribute set $(i, \omega_1), (i, \omega_2), \ldots, (i, \omega_q)$ to request the corresponding signature key. If identity $U$ is colluded, the adversary submits the verification public key directly to the challenger, and the adversary queries the challenger for the decryption key corresponding to identity $U$ and attribute set $\omega$.
- **Forge**: The adversary outputs a $(U^*, \omega^*)$ decryption key $SK_{U^*,\omega^*}$. If $\text{Trace}(PK, SK_{U^*,\omega^*}) \neq \bot$, and $SK_{U^*,\omega^*}$ does not come from the query of $(U^*, \omega^*)$ decryption key, the adversary wins the game, and its advantage is defined as $\text{Adv}(A) = \Pr[A_{\text{wins}}]$.

Definition 5: In the security game as mentioned above, if all polynomial-time adversaries have at most one negligible advantage, then the traceable CP-HABE-AKDA has white box traceability.

### D. TRACEABLE CP-HABE-AKDA CONSTRUCTION

We construct the traceable CP-HABE-AKDA scheme as follows, the workflow of which is shown in Fig.6:

- **GlobalSetup**: Given a security parameter $\lambda$ and an attribute universe $\mu$ with the size of $n$. The entire attribute set is $\mu := \{1, 2, \ldots, n\}$, where $n$ is a natural number. The algorithm first chooses two bilinear groups $G$ and $G_T$ of prime order $p$, two random generators
the user whose attribute set is $\omega$. RA broadcasts public key PK and keep the master key MK:

$$PK = (p, G, GT, e, g0, h0, g3, f1, f2, \ldots, fL, H, Y, \mu)$$

- **USignKeyGen**: The algorithm is run by the user and generates a pair of user signed keys. User $U$ randomly selects an element $x_U \in Z_p$ as the signature key and publishes $P_U = g^x_U$ as the public verified key.

- **LevelKeyGeneration**: Root authority RA assigns a unique public identity $LID_i \in B$ and a random large positive integer $y_i^e \in Z_p$ for each level, then the key parameter for each level $i$ is $k_i = H(y_i^e, LID_i)$. RA stores $k_i$ and $LID_i$ in the entry of $i$th level key table, selects a random number $r \in Z_p$, and generates the level key of domain authority $DA_i$ as follows:

$$SK_i = \left( g_0^{\beta+k_i+r} \cdot g_0^{r \cdot h_0^{+r}} \cdot (g_3 \cdot \prod_{l=1}^{i} f_l) \right)^r \cdot f_{i+1}^{k_i} \cdots f_k$$

The key $SK_i$ and key parameter $LID_i$ are sent to $DA_i$ through a secure channel between RA and $DA_i$.

- **LevelKeyDerivation**: It is assumed that the security level $i$ is the predecessor of the security level $h$, $1 \leq i < h \leq L$, and their key parameters are $k_i$ and $k_h$ respectively. $DA_i$ uses the public information preset by RA $SK_i$ to generate a new level key $SK_h$ to realize level key-delegation.

First, the root authority RA randomly selects a large positive integer $q$ to store derived public information $y_{ih} = k_h - H(k_i, LID_h) \mod q$ for the relationship $(V_i, V_h)$ in the hierarchy, which is from level $i$ to level $h$. $q$ is published by RA later.

Then, according to the public information $LID_h$ and $y_{ih}$, $DA_i$ uses its key parameter to calculate the value of key parameter $k_h$ of level $h$, that is $k_h = y_{ih} + H(k_i, LID_h) \mod q$.

Finally, $DA_i$ selects a random number $r' \in Z_p$, and uses its level key $SK_i$ to generate level key $SK_h$ related to level, as follows:

$$SK_h = \left( g_0^{\beta+k_h+r+r'} \cdot g_0^{r+r'} \cdot h_0^{a+r+r'} \cdot (g_3 \cdot \prod_{l=1}^{h} f_l) \right)^{r+r'} \cdot f_{h+1}^{k_h} \cdots f_k$$

**AttributeKeyGeneration**: The attribute private key $SK_{U,\omega}$ of the user whose attribute set is $\omega$ at level $i$ is generated by root authority RA, domain authority $DA_i$ and the user $U$ in the following steps:

**Step 1**: $DA_i$ sends $\{D_{i,\omega} \}_{i \in \mu}$ to user $U$ through the secure channel between domain authority and user.

**Step 2**: The domain authority $DA_i$ at level $i$ generates attribute private key $SK_{U,\omega}$ for user $U$ with attribute set $\omega \subseteq \mu$. $DA_i$ selects secret random number $t_1, t_2, \ldots, t_n \in Z_p$, and computes the public parameters $h_1 = h_0^0, \ldots, h_n = h_{n+1}^0, h_n = h_{n+1}^h$. In addition, $n - 1$ random values $x_1, x_2, \ldots, x_{n-1}$ are randomly chosen in $Z_p$ and associated with $k_i$ by computing $x_n = (-k_i) - x_1 - \cdots - x_{n-1} \in Z_p$. For each attribute $j \in \omega$, a random number $x_n$ is assigned as follows:

$$D_{i,\omega} = \begin{cases} g_0^j & (j \in \omega) \\ g_0^{x_n} & (j \in \mu \setminus \omega) \end{cases}$$

$DA_i$ sends $\{D_{i,\omega} \}_{i \in \mu}$ to user $U$ through the secure channel.

**Step 3**: User $U$ selects a random number $\lambda \in Z_p$, and outputs the private key of the attribute set $\omega$ at level $i$ as:

$$SK_{U,\omega} = \begin{cases} D_1 = r_U, D_2 = g_0^{(1/\lambda + r_U)} \cdot g_0^{\beta+k_i+r}, D_3 = g_0^{(1/\lambda + r_U)} \\ D_4 = (g_3 \cdot \prod_{l=1}^{i} f_l)^{r+1/\lambda} \cdot D_{i,\omega} \end{cases}_{i \in \mu}$$

- **Encryption**: This algorithm encrypts a message $M \in G_T$ under the access control policy specified with an access structure $T = \Lambda_{j \in W}$ at level $i$, where $W \subseteq \omega$. The encryption algorithm first selects a random value $s \in Z_p$. Then, the ciphertext $CT$ is composed as follows:

$$CT = \left( E' = MY^{(\alpha-\beta)s}, h_0^{s \cdot g_0^s} \cdot (g_3 \cdot \prod_{l=1}^{i} f_l)^s, \{E_{i,\omega} = h_i^t \}_{j \in W} \right)_{i \in \mu \setminus \omega}$$

- **Decryption**: There are two requirements on successful decryption: first, the attribute set associated with the user needs to satisfy the access structure; second, the attribute set needs to be at the correct level. Therefore, the decryption algorithm consists of two steps: the first step confirms whether the attribute set $\omega$ satisfies the access structure $T$; and the second step uses the attribute private key of the correct level for message recovery. The decryption algorithm uses algorithm $DecAtt(CT, SK_i_{U,\omega}, \omega)$ to execute the following steps:

**Step 1**: For all attributes $j \in \mu$, if the set $\omega \subseteq \mu$ composed of conforms to the access structure $T$, $DecAtt(CT, SK_i_{U,\omega}, \omega)$ outputs $e(g_0, h_0)^{(-k_i)}$; If such a set does not exist and the
access structure is not matched, the algorithm outputs ⊥.

\[
\text{DecAttr}(CT, SK_{i,o}) = \prod_{j \in W \cup \mu \cup \lambda} e(D_{i,o}, E_{i,o}) \prod_{j \in W} e(D_{i,o}, E'_{i,o})
\]

\[
= \prod_{j \in \omega} e\left(g_0^{y_j}, h_j^k\right) \prod_{j \in \mu \cup \lambda} e\left(g_{n+j}^{y_j}, h_{n+j}^k\right)
\]

\[
= \prod_{j \in \omega} e\left(g_0^{y_j}, h_j^k\right) \prod_{j \in \mu \cup \lambda} e\left(g_{n+j}^{y_j}, h_{n+j}^k\right)
\]

\[
= e(g_0, h_0)^{\sum y_j} = e(g_0, h_0)^{-k_i} = Z(1)
\]

**Step 2.** After confirming that the attribute set satisfies the access structure, the decryption algorithm executes:

\[
Z^{(2)} = e\left(g_0^{\beta + k_i + r}, h_0^k\right) = e(g_0, h_0)^{\beta + k_i + r}
\]

\[
Z^{(3)} = e\left(g_0^{\beta + k_i + r}, g_3 \cdot \prod_{l=1}^{\beta} f_l\right)
\]

\[
= e(g_0, h_0)^{\beta + k_i + r}
\]

\[
M = E' \cdot Z^{(1)} Z^{(2)} Z^{(3)}
\]

**Trace:** The identity tracking algorithm takes the user’s attribute private key as input, and the root authority RA checks its key element to verify whether the user’s attribute private key is valid. After verification, RA queries the user registry \(Index_{RA}\) based on the elements in the key construction and traces out the user holding the key.

First, RA checks if the user attribute private key elements make the following formulas hold:

1) \(r_U \in \mathbb{Z}_p\)

2) \(e(D_1, g_0) = e(D_2, g_3 \cdot \prod_{l=1}^{\beta} f_l) e(h_0, g_0^{\beta + r})\)

3) \(\prod_{j \in \omega} e(D_{i,o}, h_j^k) = e(D_3, g_3^{\beta + 2k_i + r}) e(h_0, g_0^{\beta + k_i + r})\)

If all the above equations are true, the user attribute private key \(SK_{U,o}\) passes the identity check, and the tracking algorithm outputs 1; otherwise, the tracking algorithm outputs 0. If the output is 1, RA finds \((r_U, F, P_U, IDinfo_U, w)\) from the user registry \(Index_{RA}\) and verifies whether \(e(F, g) = e(h_1, P_U)\) is valid, when \(h_1 = H_1(r_U, P_U, IDinfo_U)\). If it holds, RA outputs the user identity information \(IDinfo_U\) which is corresponding to \(SK_{U,o}\).

**VIII. SECURITY ANALYSIS OF TRACEABLE CP-HABE-AKDA**

**A. SECURITY PROOF OF CONSTRUCTION**

Theorem 3: The proposed traceable CP-HABE-AKDA scheme is IND-s-CPA secure with respect to Definition 4 in the generic group model.

**Proof:** The proof is similar to that for Theorem 1, and hence, we omit it.

**B. TRACEABILITY PROOF OF CONSTRUCTION**

Theorem 4: If \(l - SDH\) assumption holds, the traceable CP-HABE-AKDA scheme is white box traceable, that is, the user can not generate a forged key.

**Proof:** Suppose there exists an adversary \(A\) that has non-negligible advantage in winning the traceability game after making \(q\) key queries, without the generality, let \(q = q + 1\). A probabilistic time polynomial algorithm \(B\) can be constructed to break \(l - SDH\) assumption with the advantage of \(e\). The interaction between \(A\) and \(B\) is as follows:

- **Init:** \(B\) receives one of parameters \((\bar{g}, \bar{g}^\beta, \ldots, \bar{g}^{\beta q})\) in \(l - SDH\) assumption, where \(\bar{g}\) is an element on \(G\) and sets \(B_U = \bar{g}^{\sigma_U}, U = 0, 1, \ldots, q\). The goal of \(B\) is to output a tuple \((c_U, \sigma_U)\) in \(Z_p \times G\) that satisfies \(\sigma_U = g^{1/(\beta + c_U)}\) to solve \(l - SDH\) problem. Then algorithm \(B\) selects \(q\) random numbers \(c_1, c_2, \ldots, c_q \in Z_p^\ast\) and sets the polynomial function \(f(y)\) to the following form: \(f(y) = \prod (y + c_U)\). Make \(f(y)\) polynomial expansion and record it as \(f(y) = \sum_{\eta_U} \eta_U y^U\), where \(\eta_0, \eta_1, \ldots, \eta_q \in Z_p\) are the coefficients of the polynomial. \(B\) then calculates:

\[
g = \prod_{U=0}^{q-1} B_U^{c_U} = \prod_{U=0}^{q-1} (g^y)^{\eta_U} = g^{f(\delta)}
\]

\[
g^{\delta} = \prod_{U=1}^{q} B_U^{\eta_U-1} = \prod_{U=1}^{q} (g^y)^{\eta_U-1} = g^{f(\delta)}
\]

\(B\) creates an identity list \(T = \emptyset\) and broadcasts the public key:

\(PK = (p, G, H, e, g_0, h_0, g^3, f_1, f_2, \ldots, f_L, H, Y, \mu)\)

- **Phase1:** In the \(i\)th query, adversary \(A\) submits the parameter \((IDinfo_U, w_U)\) to \(B\), and queries the corresponding private key for the user with the attribute set \(w_U\) at level \(i\), whose identity is \(IDinfo_U\). Let the polynomial function \(f_U(y)\) be of the form:

\[
f_U(y) = f(y) + c_U = \prod_{\delta=1, \delta \neq i}^{q-1} (y + c_U)
\]

Similarly, polynomial expansion can be obtained as follows:

\[
f_U(y) = \sum_{\delta=0}^{q-2} \theta_\delta y^\delta
\]

\(B\) computes:

\[
\sigma_U = \sum_{\delta=0}^{q-2} B_\delta^{c_U} = \sum_{\delta=0}^{q-2} (g^y)^{\delta} = g^{f(\delta)} = g^{1/(\beta + c_U)}
\]

Then \(B\) randomly selects and calculates:
\[ K = D_2 = g_0^{r^+[1/(\lambda+r)])} = a_U \cdot g_0^L, L = D_3 = g_0^{g_2[r^+r]} \cdot D_4 = g_0^{g_3[r]}, D_1 = c_U = r_U \], and adds \((c_U, ID_{influence})\) to the table \(T\).

- **Forge** The adversary \(A\) submits a private key \(SK^* = \{D_1^*, D_2^*, D_3^*, \{D_{i,co}\}_{j\in J}\}\) to \(B\). According to the assumption, the private key can pass the validity verification and \(D_1^* \notin \{c_1, c_2, \ldots, c_q\}\). After receiving the private key, \(B\) first rewrites the polynomial function \(f(y)\) as:

\[
f(y) = \gamma(y) (y + D_1^*) + \gamma - 1
\]

where the remainder term \(\gamma - 1 \in \mathbb{Z}_p\), and \(\gamma\) is a polynomial about \(y\). \(\gamma(y) = \sum_{U=0}^{q-2} y_U y^U\). Divide both ends of equation \((*)\) by \((y + D_1^*)\) to get the rational fraction:

\[
\frac{f(y)}{(y + D_1^*)} = \gamma(y) + \frac{\gamma - 1}{(y + D_1^*)}
\]

Since \(f(y) = \sum_{U=1}^{q} (y + c_U)\) and \(D_1^* \notin \{c_1, c_2, \ldots, c_q\}\), we can know from the basic algebra theorem that \((y + D_1^*)\) cannot divide \(f(y)\), so the remaining terms \(\gamma - 1 \neq 0\). Then \(B\) computes \(1/\gamma - 1 (mod p)\) (since \(gcd(\gamma - 1, p) = 1\)) and then does the following computation:

\[
\sigma^* \leftarrow D_2^* (D_3^*)^\sigma = g_8^{\frac{1}{1+y^p}} \cdot g_8^{(l_k)} \cdot (g_8)^{y(\lambda)} \cdot (g_8)^{\gamma - 1} = \prod_{i=0}^{q-1} B_i \cdot g_8^{\frac{1}{1+y^p}}
\]

\[\sigma^* = e \left( \tilde{g}_8, g_8^* \right) \mod p \in \mathbb{Z}_p\]

Since \(e \left( \tilde{g}_8, g_8^* \right) = e \left( \tilde{g}_8, g_8^{D_1^*} \cdot g_8^{\frac{1}{1+y^p}} \right) = e \left( \tilde{g}_8, g_8 \right)\), the solution of \(l - SDH\) problem is \(D_1^*, \sigma^*\).

Now we evaluate the advantage of \(B\) in breaking \(l - SDH\) assumption.

Let \(\ell_{SDH}\) denote the event that \((D_1^*, \sigma^*)\) is a solution for the \(l - SDH\) problem, which can be verified by checking whether \(e \left( \tilde{g}_8, g_8^*, \sigma^* \right) = e \left( \tilde{g}_8, g_8 \right)\) holds. Note that when \(B\) randomly chooses \((D_1^*, \sigma^*)\), \(\ell_{SDH}\) happens with negligible probability, mark zero for simplicity. Moreover, the probability of the event happens when \((D_1^*, \sigma^*)\) satisfies \(e \left( \tilde{g}_8, g_8^*, \sigma^* \right) = e \left( \tilde{g}_8, g_8 \right)\) is 1 in the case of \((Aw_{in} \land gcd(\gamma - 1, p))\) when \(B\) outputs \((D_1^*, \sigma^*)\). So the probability for \(B\) has a solution of \(l - SDH\) problem is calculated as follows:

\[
Pr \left[ \ell_{SDH} \left( e^*, \sigma^* \right) \right] = \Pr \left[ \ell_{SDH} \left( e^*, \sigma^* \right) | Aw \right] \cdot \Pr \left[ Aw \right]
\]

\[+ \Pr \left[ \ell_{SDH} \left( e^*, \sigma^* \right) | Aw \land gcd(\gamma - 1, p) \neq 1 \right] \cdot \Pr \left[ Aw \land gcd(\gamma - 1, p) \neq 1 \right] + \Pr \left[ e^* \right] \cdot \Pr \left[ Aw \land gcd(\gamma - 1, p) = 1 \right]
\]

\[= \Pr \left[ Aw \right] \cdot \Pr \left[ gcd(\gamma - 1, p) = 1 \right] = \varepsilon
\]

Therefore, the algorithm \(B\) can solve the \(l - SDH\) problem with non-negligible advantage, which conflicts with the \(l - SDH\) assumption, so the theorem proves that the proposed traceable CP-HABE-AKDA scheme has white-box traceability.

**IX. PERFORMANCE ANALYSIS**

In terms of performance analysis, factors that affect the performance of the proposed scheme are taken into account, mainly considering the size of key storage space and the cost of attribute set-related algorithms. In this section, from the aspect of storage overhead, the length of user attribute private key is used as an indicator; from the aspect of computational overhead, the user attribute private key is generated, and the amount of encryption and decryption operations are used as indicators. The proposed scheme is compared with the HABE scheme represented by the Wang scheme [20], Wan scheme [21] and Asim scheme [7]. For convenience, the notations used in the following description is introduced in Table 3.

In the proposed schemes, level key generation and encryption operation are related to the computing cost of root authority RA, attribute private key generation is related to the computing cost of domain authority DA, and encryption...
operation is related to the computing cost of the user. The computational overhead of several schemes is shown in Table 4.

In Wang’s scheme [20], a file is encrypted based on the Disjunctive Normal Form (DNF) access control strategy \( A = \bigvee_{i=1}^{N} (CC_i) \), and the user needs to calculate \( O(1) \) bilinear mappings and \( O(N_TL) \) exponentiation operations to output a ciphertext of length \( O(N_T) \). \( L \) is the maximum depth of managed attributes in all domain hosts. The bilinear operation is independent of the encrypted message so that it can be done at once. In order to restore the file, the user needs to perform \( O(1) \) bilinear mapping operations with the private key of the user attribute with the key length \( 1 + M + |A_u| \).

When Wan scheme [21] generates the key of length \( 2|A_u| + |A_m| \) for the user, the algorithm and mechanism adopted are the same as that used for generating the key of the domain. The cost is determined according to the number of attributes owned by the user and the number of subsets in the recursive attribute set. The computational complexity is \( O(2n + A_m) \). The encryption algorithm adopts the form of an access control tree. The transformation nodes of the leaf node set in the access control tree determine the encryption cost. The tree access structure \( \tau \) is used to encrypt DEK, including 2 exponentiation operations for each leaf node in \( \tau \) and 1 exponentiation operation for each transformation node. Therefore, the computational complexity of the new file creation is \( O(2 |N_y| + |N_x|) \), where \( y \) represents the number of leaf nodes in \( \tau \) and \( x \) represents the transformation nodes in \( \tau \). Since the decryption overhead depends on the access control tree and key structure, the decryption operation consists of two bilinear pairings for each leaf node on the path from leaf node to root node, and one exponentiation operation for each node on the path from leaf node to root node.

When the Asim scheme [7] generates an attribute private key of length \( |A_u| + |A_m| \) for the user, the overhead is determined according to the number of attributes owned by the user and the level of the user. A total of \( i + 5 + 2n \) exponentiation operations are used, where \( i \) represents the level of the user and \( n \) represents the number of attributes in the user attribute set. When encrypting, the form of access control tree is used. The encryption cost is determined by the number of nodes and levels in the access control tree adopted by the encryption algorithm, so the overhead of file encryption includes \( i + 2 + 2n \) exponentiation operations and one bilinear pairing operation. The cost of the decryption operation depends on the user attribute private key. The decryption operation includes three bilinear pair operations to satisfy the leaf nodes of the access tree and one bilinear pair for each transition node on the path from the leaf node to the root node, and \( 2i \) exponentiation operation used for the attribute set at level \( i \).

The proposed schemes take each attribute in AND-gate structure as a unit to realize the user attribute private key generation, encryption, and decryption. Therefore, the length of the user attribute private key is related to \( n \), which is the number of total attributes, and linearly increases with the universe of attributes. In the generation of the user attribute private key, the level key parameter and each attribute in the attribute set are embedded by exponentiation operation. Through bilinear mapping, the decryption qualification of the attribute private key is verified, and then the decryption is completed. Since the calculation cost of Wan scheme [21] and Asim scheme [7] is related to the number of nodes in an access tree structure, the more complex the tree structure is, the more computation is required. However, the calculation cost of the proposed schemes is related to the size of the entire attribute set. It has nothing to do with the internal structure. Therefore, when the total attribute set is fixed, the proposed scheme is more efficient.

### X. CONCLUSION

In this paper, we focus on the issue of key-delegation in the HABE scheme and propose the key-delegation abuse problem in CP-HABE, which could severely affect the data security when adopted in IoT-connected healthcare system. In order to solve the problem, we summarize two causes of key-delegation abuse in CP-HABE and present a targeted CP-HABE-AKDA solution. In the proposed scheme, the directed graph is applied to realize the key-delegation between domain authorities from the upper level to the lower level. By embedding the level key parameter into a user attribute key, our construction can prevent the domain

| Schemes | Access Structure | The length of attribute private key | The generation of attribute private key | Encryption | Decryption |
|---------|-----------------|-----------------------------------|----------------------------------------|------------|-----------|
| Wang[20] | DNF             | \( 1+M+|A_u| \)             | \( 1+M+|A_u| \)                        | \( T_{pair} + N_T L T_e \) | \( T_{pair} \) |
| Wan[21]  | AND-gate/Access tree | \( 2|A_u| + |A_m| \) | \( (2n + A_m) T_{exp} \) | \( (2N_y + N_x) T_e \) | \( (2N_y + N_x) T_{pair} + N_p T_e \) |
| Asim[7]  | Access tree     | \( |A_u| + |A_m| \)              | \( (i + 5 + 2A_u) T_e \)                      | \( (i + 2 + 2A_u) T_e + T_{pair} \) | \( (3N_y + N_x + 8) T_{pair} + (2i + 4) T_e \) |
| Ours 1   | AND-gate       | \( |A_u| + n \)                  | \( (n + 5) T_e \)                          | \( (n + 2) T_e + T_{pair} \)       | \( (n + 4) T_{pair} \) |
| Ours 2   | AND-gate       | \( |A_u| + n \)                  | \( (n + 4) T_e \)                          | \( (n + 2) T_e + T_{pair} \)       | \( (n + 7) T_{pair} \) |
authority from creating decryption keys for users who do not comply with the access policy. We further study the strategy of tracking malicious user’s identity when key leakage happens, especially in IoT-connected healthcare system, and designs a traceable CP-HABE-AKDA solution that can resist key-delegation abuse and has white-box traceability. Based on the generic group model and the random oracle model, we demonstrate the CPA security of the proposed schemes and the ability to resist key-delegation abuse attacks. Moreover, we complete the traceability proof of the proposed traceable CP-HABE-AKDA scheme based on $I – SDH$ assumption.

As mentioned above, the key-delegation abuse not only exists in the HABE scheme with AND-gate access structure, but also in the schemes with the access tree and LSSS structure. The solutions provided in this paper only considers the case of AND-gate. In practice, especially in the IoT-connected healthcare system, HABE schemes are mainly with the access tree structure. Therefore, in the next stage of work, we continue to expand the HABE schemes against key-delegation abuse to the schemes with access tree and LSSS structure. Furthermore, supporting user revocation for the proposed scheme with traceability is also a potential research work we will explore.

REFERENCES

[1] L. A. Tawalbeh, R. Mehmood, E. Benkhlifa, and H. Song, “Mobile cloud computing model and big data analysis for healthcare applications,” *IEEE Access*, vol. 4, pp. 6171–6180, 2016.

[2] Z. Zhang, W. Zhang, and F.-H. Tseng, “Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques,” *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan. 2019.

[3] M. M. Dhanvijay and S. C. Patil, “Internet of Things: A survey of enabling technologies in healthcare and its applications,” *Comput. Netw.*, vol. 153, pp. 113–131, Apr. 2019.

[4] 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), 2006, pp. 89–98.

[5] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[6] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in *Proc. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2011, pp. 53–70.

[7] M. Asim, T. Ignatenko, M. Petkovic, D. Trivellato, and N. Zannone, “Enforcing access control in virtual organizations using hierarchical attribute-based encryption,” in *Proc. 7th Int. Conf. Availability, Rel. Secur. (ARES)*, Prague, Czech Republic, Aug. 2012, pp. 212–217.

[8] H. Deng, Q. Wu, B. Qin, J. Domingo-Ferrer, L. Zhang, J. Liu, and W. Shi, “Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts,” *Inf. Sci.*, vol. 275, pp. 370–384, Aug. 2014.

[9] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, “Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing,” *Future Gener. Comput. Syst.*, vol. 78, pp. 720–729, Jan. 2018.

[10] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Proc. Ann. Int. Conf. Theory Appl. Cryptograph.* Techn. Berlin, Germany: Springer, 2005, pp. 457–473.

[11] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based encryption with non-monotonic access structures,” in *Proc. 14th ACM Conf. Comput. Commun. Secur.* (CCS), 2007, pp. 195–203.

[12] L. Cheung and C. Newport, “Provably secure ciphertext policy ABE,” in *Proc. 14th ACM Conf. Comput. Commun. Secur.* (CCS), 2007, pp. 456–465.

[13] V. Goyal, A. Jain, O. Pandey, and A. Sahai, “Bounded ciphertext policy attribute based encryption,” in *Proc. Int. Colloq. Automat. Lang., Program.*, in Lecture Notes in Computer Science, vol. 5126, 2008, pp. 479–591.

[14] L. Li, T. Gu, L. Chang, Z. Xu, Y. Liu, and J. Qian, “A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram,” *IEEE Access*, vol. 5, pp. 1137–1145, 2017.

[15] H. Qiao, J. Ren, Z. Wang, B. Ha, and H. Zhou, “Compulsory traceable ciphertext-policy attribute-based encryption against privilege abuse in fog computing,” *Future Gener. Comput. Syst.*, vol. 88, pp. 107–116, Nov. 2018.

[16] G. Yu, Y. Wang, Z. Cao, J. Liu, and X. Wang, “Traceable and undeniable ciphertext-policy attribute-based encryption for cloud storage service,” *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, pp. 1–10, Apr. 2019.

[17] R. Li, H. Asaeda, and J. Li, “A distributed publisher-driven secure data sharing scheme for information-centric IoT,” *IEEE Internet Things J.*, vol. 4, no. 3, pp. 791–803, Jun. 2017.

[18] J. Li, Q. Wang, C. Wang, and K. Ren, “Enhancing attribute-based encryption with attribute hierarchy,” *Mobile Netw. Appl.*, vol. 16, no. 5, pp. 553–561, Oct. 2011.

[19] 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.* (CCS), Chicago, IL, USA, 2010, pp. 735–737.

[20] G. Wang, Q. Liu, J. Wu, and M. Guo, “Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers,” *Comput. Secur.*, vol. 30, no. 5, pp. 320–331, 2011.

[21] 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.

[22] 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.

[23] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, “Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption,” in *Advances in Cryptology* ( Lecture Notes in Computer Science, vol. 6110, H. Gilbert, Ed. Berlin, Germany: Springer, 2010, pp. 62–91.

[24] A. Lewko and B. Waters, “New techniques for dual system encryption and fully secure HIBE with short ciphertexts,” in *Proc. 7th Theory Cryptogr. Conf.*, Zürich, Switzerland, Feb. 2010, pp. 455–479.

[25] A. Lewko, Y. Rouselakis, and B. Waters, “Achieving leakage resilience through dual system encryption,” in *Proc. 8th Theory Cryptogr. Conf.*, Providence, RI, USA, Mar. 2011, pp. 70–88.

[26] S. Wu, 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.

[27] J. Li, Q. Yu, and Y. Zhang, “Hierarchical attribute based encryption with continuous leakage-resilience,” *Inf. Sci.*, vol. 484, pp. 113–134, May 2019.

[28] Q. Huang, L. Wang, and Y. Yang, “DECENT: Secure and fine-grained data access control with policy updating for constrained IoT devices,” *World Wide Web*, vol. 21, no. 1, pp. 151–167, Jan. 2018.

[29] Y. Wang and J. Gao, “A regulation scheme based on the ciphertext-policy hierarchical attribute-based encryption in bitcoin system,” *IEEE Access*, vol. 6, pp. 16267–16278, 2018.

[30] J. Sun, S. Hu, and X. Nie, “Fine-grained ranked multi-keyword search over hierarchical data for IoT-oriented health system,” *IEEE Access*, vol. 7, pp. 101969–101980, 2019.

[31] S.-Y. Tan, K.-W. Yeow, and S. O. Hwang, “Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6384–6395, Aug. 2019.

[32] J. Herranz, F. Laguillaumie, and C. Ráfols, “Constant size ciphertexts in threshold attribute-based encryption,” in *Public Key Cryptography*. Berlin, Germany: Springer, 2010, pp. 19–34.

[33] V. Shoup, “Lower bounds for discrete logarithms and related problems,” in *Proc. Int. Conf. Theory Appl. Cryptograph.* Techn. Berlin, Germany: Springer, 1997, pp. 256–266.

[34] D. Boneh and X. Boyen, “Short signatures without random oracles,” in *Advances in Cryptology* ( Lecture Notes in Computer Science, vol. 3027. Berlin, Germany: Springer-Verlag, 2004, pp. 56–73.
XI CHEN is currently pursuing the Ph.D. degree with the Department of Information and Communication Engineering, Beijing Jiaotong University. Her research interests include cryptographic protocols and information security.

HAN-CHIEH CHAO (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1993, respectively. He is currently a Professor with the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan, where he also serves as the President. He is also with the Department of Computer Science and Information Engineering and the Department of Electronic Engineering, National Ilan University, Yilan, Taiwan, the College of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan, China, and the Fujian University of Technology, Fuzhou, China. He serves as the Editor-in-Chief for the Institution of Engineering and Technology Networks, the Journal of Internet Technology, the International Journal of Internet Protocol Technology, and the International Journal of Ad Hoc and Ubiquitous Computing. He is a Fellow of the Institution of Engineering and Technology (Institution of Electrical Engineers) and a Chartered Fellow of the British Computer Society.

YUN LIU received the Ph.D. degree in electronic engineering from Beijing Jiaotong University, in China. She is currently a Professor, the Director of the Department of Communication and Information System, and the Director of the Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University. She has edited many books and published more than 200 articles and book chapters, as well as participated in many international conferences. Her current research interests include telecommunication, computer networks, network security, intelligent transportation systems, and social dynamics. She is an IET Fellow.

YONG LI received the M.S. degree in computer science from Wuhan University, in 2003, and the Ph.D. degree from the State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, in 2007.

He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. He has more than 40 publications and filed six patents. His research interests include cryptographic protocols, cyberspace security, cloud computing security, big data security, and blockchain.

Dr. Li is a member of the International Association for Cryptologic Research (IACR), the Chinese Association for Cryptologic Research (CACR), the Association for Computing Machinery (ACM), and the China Computer Federation (CCF).