

# A Hidden Tree Structure CP-ABE with Decryption Testing Scheme and its Application in Cloud Computing

Yang Zhao · Hu Xiong

Received: date / Accepted: date

**Abstract** With the rapid increase of data and the tremendous improvement of cloud computing infrastructures, more and more users are storing their personal data(e.g electronic medical records, address books, smart phone data etc.) in cloud, but the security of cloud storage supported by cloud service providers(CSP) is semi-secure to users, and the privacy of users cannot be fully guaranteed. Therefore, how to protect the security and confidentiality of data in cloud storage has become hot spot that researchers pay attention to. Since attribute-based encryption has the characteristics of one-to-many encryption and fine-grained access control, attribute-based encryption scheme is widely considered as an ideal solution. However, the basic CP-ABE encryption scheme has the flaws that access structure may leak user privacy, inefficiency, and weak expression ability. In view of the above shortcomings, this paper proposes a CP-ABE encryption scheme with the following features: 1. in order to enhance the express ability the access structure, we adopt a tree-based access structure instead of the widespread AND gate structure; 2. a testing phase is used to eliminate unnecessary operations and improve decryption efficiency; 3. to fully utilize the resources of the cloud server and decrease the local decryption options, we outsource the time-consuming pairing operation to security cloud server. Besides, the scheme is provably secure against adaptively chosen-ciphertext attack. Finally, we apply our scheme into EMR system established in cloud computing environment.

---

Yang Zhao

University of Electronic Science and Technology of China, Chengdu, 610054 , China.

E-mail: zhaoyang@uestc.edu.cn

✉ Hu Xiong

University of Electronic Science and Technology of China, Chengdu, 610054 , China.

Guangxi Colleges and Universities Key Laboratory of cloud computing and complex systems, Guilin University of Electronic Technology, Guilin, 541004, China.

E-mail: xionghu.uestc@gmail.com

**Keywords** ciphertext policy attribute-based encryption; hidden policy; cloud computing;

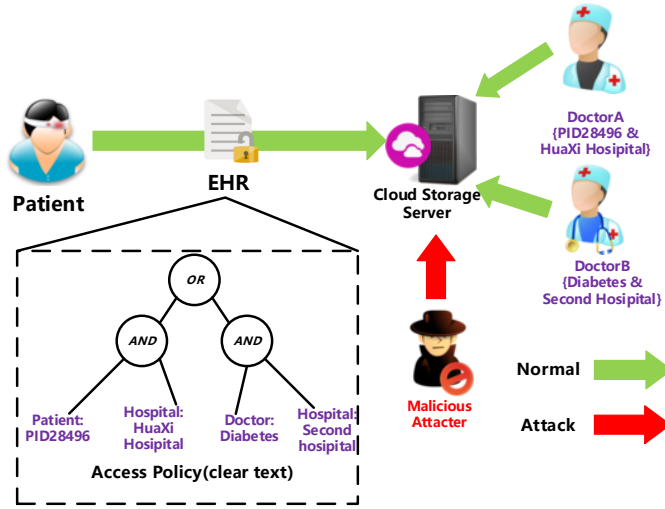
## 1 Introduction

In the era of data explosion, the development of cloud computing has given users a better and more convenient way to share data and store data. Users can store and retrieve their own data on the cloud platform and also easily share data with other users on the cloud platform. However, since the user stores the data on the cloud platform and the data is out of the user's control, it will bring out an apparent issue that if malicious users attack cloud server and obtain the data belonging to normal users, the user's privacy will be leaked and even causes incalculable losses. Therefore, how to ensure the privacy of users and the security of cloud data has become an urgent problem to be solved. ABE as a novel encryption notation, not only provides access control but also provides the function of encrypting data, consequently, it is regarded as a positive method to handle this matter. However, although attribute-based encryption can solve above problems, it has another hidden danger which is access structure is passed in clear text during the process of encryption and decryption. Once some malicious attackers steal the access structure stored in clear text, the user's privacy will be revealed.

Figure1 is a simple Electronic Medical Record System structure. Here we take the EMR system as an example. For convenience, we simplify EMR system to three parts: data owner(including patients), cloud storage server(CSS), data recipient(including doctor, malicious attacker). The data owner encrypt his EHR via CP-ABE and upload it to CSS. CSS are deployed in an open environment that anyone can access. Under normal circumstances, a data recipient who has the private key can decrypt ciphertext and obtain EMR transmitted by data owner. For example, both doctorA and doctorB in Figure1 can normally decrypt the EMR. In another case, for a malicious attacker, although he can not decrypt the EMR, he is able to get the access structure stored in clear text and obtain the data owner's information. For instance, in Figure1, the malicious may get the sensitive information that patient whose *PID* is 28496 suffers from diabetes.

### 1.1 Related Work

In 1984, Shamir introduced the idea of identity-based encryption(IBE) to address certificate management and update issues in public key cryptography. In the scheme, the user's identity is used as the public key, and the public key is allowed to be an arbitrary string, such as an email address. However, IBE can only achieve one-to-one decryption, which means that the owner encrypts the message through the recipient's public key, and the message can only be decrypted by the receiver, which is coarse-grained access control. Based on IBE,



**Fig. 1** EMR Structure

Sahai and Waters proposed a fuzzy identity-based encryption scheme (Fuzzy IBE). In the Fuzzy IBE, the user has multiple identity attributes, and only the decryptor needs to satisfy the threshold  $d$  identity features to decrypt correctly. This is the original Attribute-Based Encryption (ABE) mechanism. In 2006, Goyal et al. introduced KP-ABE, a crypto system for fine-grained sharing of encrypted data. In this scheme, the ciphertext is formed with the set of attributes and user key is embedded with policy. When the user attributes satisfy the access structure the user can decrypt the message. In 2007, Ostrovsky et al. introduced a non-monotonic access structure containing positive and negative attributes. It supports AND, OR and NOT between attributes, enriching the logical expression ability of access structure, and filling the gaps that cannot represent logical Non. Cheung et al. improved the security proof of Bethencourt et al. work and proved for the first time that the chosen ciphertext under the standard model and the Decision Bilinear Diffie-Hellman (DBDH) assumption is secure. In 2008, Goyal et al. proposed an improved version of CP-ABE that supports the bounded size access trees with threshold gate and security proved the under the standard model, decisional Bilinear Diffie-Hellman assumption. In 2011, Waters proposed a new approach to express access control through a linear secret sharing scheme (LSSS) matrix over attributes. However, the size of the ciphertext and the encryption and decryption overhead increase linearly.

In CP-ABE scheme, when a message sender encrypts a message, along with the encrypted message, the sender species in clear text the policy used to encrypt the message. It will cause a serious situation that the policy may reveal some sensitive information about the message being encrypted. Therefore,

hidden access structure in CP-ABE scheme is also crucial. In 2008, Nishide et al. proposed a partially hidden CP-ABE scheme to overcome the above problems. They used AND gate on multi-valued attributes with wildcard access structures and used inner product predicate encryption techniques, and proved that their scheme is selectively secure. In 2011, Lai et al. improved the Nishide et al. scheme and proposed a fully secure hidden policy scheme with the same access structure used in Nishide et al. The problem with this approach is that the size of the ciphertext grows with the number of attributes and leads to higher computational costs. In 2012, Li et al. used dual-system cryptography to achieve a completely secure access structure hiding scheme over a composite order group. In 2013, Hur proposed an attribute-based encryption scheme for hidden policy of the smart grid. Under the premise of hiding the access structure, a scheme for implementing arbitrary monotonic expressions is presented. But the weak points are that the user has a long length of secret key in the scheme, and no security proof is given. In 2015, Xu et al. proposed a CP-ABE scheme that supports access structure hiding based on the tree access structure, which further improved the expressiveness of the access strategy but efficiency is the bottleneck of the scheme. In 2015, Yadav et al. gave three states for each attribute in the access structure and calculated its corresponding user secret key and ciphertext to achieve a fully hidden access policy. However, the scheme only supports "AND" gate, and the expression ability is relatively weak. In 2016, Phuong et al. proposed a new scheme to overcome the ciphertext size problem in hidden strategies. However, the cumbersome user private key calculation process and complex encryption and decryption operations limit its application.

Through the ABE encryption scheme mentioned above, we can find that although they all have the identical function of hidden policy, there is a problem that cannot be neglected in terms of efficiency. The efficiency of scheme is a key points we have to consider when scheme is applied in actual scene. Moreover, most of the schemes use the access structure with AND gate, which cannot indicate threshold value, and the expression ability is lower compared with the LSSS and the tree-based structure. Finally, as mentioned above, due to the limitation of efficiency, most of the solutions do not propose a reasonable application scenario.

## 1.2 Our Contribution

User privacy-preserving and the efficiency of the solution prompt us to propose a new CP-ABE solution. In order to balance expression ability of access structure, we adopted a tree-based access structure in terms of policy hiding. For the computational complexity of the pairing operation, we make reasonable use of the advantages of cloud computing, outsourcing the decryption operation to the cloud computing server with stronger computing performance, so as to reduce the user's local calculation. In addition, during the outsourcing calculation phase, we added an additional testing step (see the implementation

section for details) to further reduce the amount of calculation. As numerous previous schemes, we also provide the proof which indicates the scheme meets Chosen Ciphertext Attack(CCA) security of the security of our scheme. Further, we depicted an EMR system architecture and smoothly embedded our solution in it. Ultimately, we demonstrate the efficiency of our scheme from both theoretical analysis and simulation experiments.

### 1.3 Organization

The remainder of this article consists of the followings. In section2, we list some preliminaries that are used in our scheme. Then, the concrete scheme and an EMR system are entirely demonstrated in section3. Section4 proposes the security proof of our scheme. Finally, in section5, the performance analysis of our scheme is illustrated.

## 2 Preliminaries

### 2.1 Access Structure

The same access structure as [??] we adopt. Let  $P_1, P_2, \dots, P_n$  be a set of parties. A collection  $\mathbb{A} \subseteq 2^{\{P_1, P_2, \dots, P_n\}}$  is monotone if  $\forall B, C : \text{if } B \in \mathbb{A} \text{ and } B \subseteq C \text{ then } C \in \mathbb{A}$ . An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection)  $\mathbb{A}$  of non-empty subsets of  $\{P_1, P_2, \dots, P_n\}$ , i.e.,  $\mathbb{A} \subseteq 2^{\{P_1, P_2, \dots, P_n\}} \setminus \{\emptyset\}$ . The sets in  $\mathbb{A}$  are called the authorized sets, and the sets not in  $\mathbb{A}$  are called the unauthorized sets.

In our scheme, the  $P_i$  are replaced by attributes. We define the collection  $\mathbb{A} = \{a_1, a_2, \dots, a_n\}$  contains all of the authorized attributes in system where  $n$  is the number of authorized attributes. For  $\forall i \in \mathbb{Z}_N$ , the value set of attribute  $a_i$  is  $\mathbb{V}_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,n_i}\}$ , similarly,  $n_i$  is the amount of elements of  $\mathbb{V}_i$ . The collection  $\mathbb{L} = \{l_1 = v_{1,t_1}, l_2 = v_{2,t_2}, \dots, l_k = v_{k,t_k}\}$  is a subset of system attributes set  $\mathbb{A}$  and it is used to denote the collection of user attributes where  $k$  is the number of user attributes.

### 2.2 Composite Order Bilinear Groups

This paper utilizes a bilinear group with a product of three unique prime numbers. Assume that the group generation algorithm  $\mathcal{G}$  has a security input parameter  $1^\lambda$  and it outputs a tuple  $(p, q, r, \mathbb{G}, \mathbb{G}_T, e)$  where  $p, q, r$  are distinct primes ( $p, q, r > 2^\lambda$ ),  $\mathbb{G}$  and  $\mathbb{G}_T$  are cyclic groups of order  $N = pqr$ .  $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r$  whose order are  $p, q, r$  separately are the subgroups of group  $\mathbb{G}$ . The  $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$  is a map with the following requirements:

- Bilinear:  $\forall g, h \in \mathbb{G}, \forall a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$ ;
- Non-degenerate:  $\exists g \in \mathbb{G}$  such that  $e(g, g)$  has the order  $N$  in  $\mathbb{G}_T$ ;

- Computable:  $\mathbb{G}$  and  $\mathbb{G}_T$  are bilinear groups if the group operations in  $\mathbb{G}, \mathbb{G}_T$  and the bilinear map  $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$  are both computable in polynomial time.

We use  $\mathbb{G}_p$  and  $\mathbb{G}_q$  to denote the subgroups of  $\mathbb{G}$  with order  $p$  and  $q$  respectively. Note that the following two properties:

- 1).  $\forall h_p \in \mathbb{G}_p, \forall h_q \in \mathbb{G}_q, e(h_p, h_q) = 1;$
- 2).  $\forall h_p \in \mathbb{G}_p, \forall h_q \in \mathbb{G}_q, \forall a, b, c, d \in \mathbb{Z}_N,$   
 $e(h_p^a h_q^b, h_p^c h_q^d) = e(h_p, h_q)^{ab} = e(h_p, h_q)^{cd}.$

### 2.3 Complexity Assumptions

Now we cite the complexity assumptions we use in this paper. The assumption is just the subgroup decision problem in the case where the group order is the product of three prime orders. Note that the assumptions have two characteristics, non-interactive and fixed size like in ???. Let  $\mathbb{G}_{pq}$  indicates the subgroup with order  $pq$ .

**Assumption 1** We define the following distributions:

$$\mathbb{G} = (N = pqr, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), g_p \xleftarrow{R} \mathbb{G}_p, g_r \xleftarrow{R} \mathbb{G}_r, D = (\mathbb{G}, g_p, g_r), \\ T_1 \xleftarrow{R} \mathbb{G}_{pq}, T_2 \xleftarrow{R} \mathbb{G}_p$$

**Definition 1**

We define the advantage of algorithm  $\mathcal{A}$  to break Assumption1 as:  $Adv_{g, \mathcal{A}(\lambda)}^1 = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|$ .

**Assumption 2** We define the following distributions:

$$\mathbb{G} = (N = pqr, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), g_p, X_1 \xleftarrow{R} \mathbb{G}_p, X_2 \xleftarrow{R} \mathbb{G}_q, g_r \xleftarrow{R} \mathbb{G}_r, D = (\mathbb{G}, g_p, g_r, X_1, X_2), \\ T_1 \xleftarrow{R} \mathbb{G}_{pq}, T_2 \xleftarrow{R} \mathbb{G}_p$$

**Definition 2**

We define the advantage of algorithm  $\mathcal{A}$  to break Assumption2 as:  $Adv_{g, \mathcal{A}(\lambda)}^2 = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|$ .

**Assumption 3** We define the following distributions:

$$\mathbb{G} = (N = pqr, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), \omega, s \in \mathbb{Z}_N, g_p \xleftarrow{R} \mathbb{G}_p, X_2, Y, Z_2 \xleftarrow{R} \mathbb{G}_q, g_r \xleftarrow{R} \mathbb{G}_r, D = (\mathbb{G}, g_p, g_p^\omega X_2, g_p^s Y_2, Z_1 Z_2, g_r), \\ T_1 \xleftarrow{R} e(g_p, g_p)^{\omega s}, T_2 \xleftarrow{R} \mathbb{G}_p$$

**Definition 3**

We define the advantage of algorithm  $\mathcal{A}$  to break Assumption3 as:  $Adv_{g, \mathcal{A}(\lambda)}^3 = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|$ .

**Definition 4** If algorithm  $\mathcal{A}$ ,  $Adv_{g, \mathcal{A}(\lambda)}^1, Adv_{g, \mathcal{A}(\lambda)}^2, Adv_{g, \mathcal{A}(\lambda)}^3$  can be ignored in any polynomial time, the  $\mathcal{G}$  will satisfy Assumption1~3.

## 2.4 Security Model

We will give a adaptively-CCA security model for our scheme. The security game between adversary  $\mathcal{A}$  and Challenger  $\mathcal{C}$  is described as followings:

- **Setup:** The challenger  $\mathcal{C}$  runs initialization algorithm  $Setup(1^\lambda)$  to generates public key  $PK$  and master secret key  $MSK$ . Then  $\mathcal{C}$  stores  $MSK$  and transfers  $PK$  to  $\mathcal{A}$ .
- **Phase1:**  $\mathcal{A}$  asks  $\mathcal{C}$  secret key of attribute sets  $L$  adaptively.  $\mathcal{C}$  generates the secret key  $SK_L$  by running  $KeyGen$  algorithm and transfers  $SK_L$  to  $\mathbb{A}$ . Note that  $\mathbb{A}$  can repeatedly ask the secret key multiple times.
- **Challenge:**  $\mathcal{A}$  submits two messages  $m_0, m_1$  which they have the same length and two access structures  $T_0, T_1$ . Note that any attribute set isn't satisfied with  $T_0$  and  $T_1$ .  $\mathcal{C}$  flips a random coin  $b \in \{0, 1\}$  and then calculates  $c^* = \text{Encrypt}(PK, m_b, T_b)$ . Finally,  $\mathcal{C}$  sends  $c^*$  to  $\mathcal{A}$  as challenge ciphertext.
- **Phase2:** Phase2 is the same as Phase1.
- **Guess:** The  $\mathcal{A}$  outputs a guess  $b'$  of  $b$  where  $b' \in \{0, 1\}$ .

If  $b' = b$ , the adversary wins this security game. And the advantage of adversary in this security game is defined as  $Adv_{\mathcal{A}} = |Pr[b' = b] - \frac{1}{2}|$ .

## 2.5 CP-ABE Scheme

The ciphertext policy attribute-based encryption (CP-ABE) scheme consists of four fundamental algorithms: Setup, Encrypt, Key Generation and Decrypt.

- $\text{Setup}(k)$ . The setup algorithm takes no input other than the security parameter  $k$ . It outputs the public parameters  $PK$  and a master key  $MK$ .
- $\text{Key Generation}(MK, S)$ . The key generation algorithm takes the master key  $MK$  and a set of attributes  $S$  that describe the key as input. It outputs a private key  $SK$ .
- $\text{Encrypt}(PK, M, A)$ . The encryption algorithm takes the public parameters  $PK$ , a message  $M$  and an access structure  $A$  over the universe of attributes as input. The algorithm will encrypt  $M$  and produce a ciphertext  $C_T$ , so that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains  $A$ .
- $\text{Decrypt}(PK, C_T, SK)$ . The decryption algorithm takes the public parameters  $PK$ , a ciphertext  $C_T$  which contains an access policy  $A$ , and a private key  $SK$  as input. If the attributes set satisfies the access structure  $A$ , the algorithm will decrypt the ciphertext and return a message  $M$ , otherwise return the error symbol.

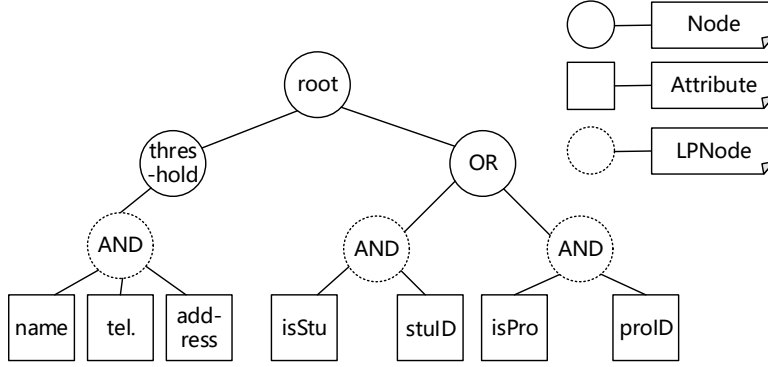


Fig. 2 Access Tree

### 3 Our Construction

In this section, we will introduce our scheme in detail. A novel access structure based on tree structure is presented firstly and then the main scheme will be described.

#### 3.1 Access Tree

To enhance the expression capability of access structure, we utilize tree structure in our scheme. Let  $\mathcal{T}$  represents a hierarchical tree. The inner nodes of  $\mathcal{T}$  are described the logic relationship of different access policies, including *AND*, *OR*, *Threshold*. Besides, the leaf nodes of  $\mathcal{T}$  represent attribute marked  $a_i = v_{i,t_i} (i \in \mathbb{Z}_n, t_i \in \mathbb{Z}_{n_i})$ . For each leaf node, we can always seek its parent node and called it *LPNode* for simplicity. In our tree structure *LPNode* can only express the *AND* relationship. Hence, it is obvious that a kind of attribute value can only appear once under each *LPNode*. Here is an access tree example in Figure2.

#### 3.2 Our Scheme

Our scheme is consisted of the following algorithm:

- $\text{Setup}(1^\lambda)$ : TA runs algorithm  $\mathcal{G}(1^\lambda)$  to product four elements including  $N = p \cdot q \cdot r, \mathbb{G}, \mathbb{G}_T, e$ . The subgroups  $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r$  are from group  $\mathbb{G}$ . Note that, the subgroup  $\mathbb{G}_p$  is used to subsequent steps. Considering security of the whole scheme, the subgroup  $\mathbb{G}_r$  is used to generate random parameters. The subgroup  $\mathbb{G}_q$  is mainly used to security analysis. And  $g_p, g_q, g_r$  are the generators of three subgroups  $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r$ .  $\mathbb{U} = \{a_1, a_2, \dots, a_n\}$  expresses all the attributes in the whole system. For each attribute, TA randomly



chooses  $a_{i,t_i} \in \mathbb{Z}_N$ ,  $R_{i,t_i} \in \mathbb{Z}_n$  and calculates  $A_{i,t_i} = g_p^{a_{i,t_i}} R_{i,t_i}$ . In addition, TA generates three more parameters  $\omega, \bar{\omega} \in \mathbb{Z}_N/0$  and  $R_0 \in \mathbb{G}_r$ . Last, TA computes  $A_0 = g_p R_0$ ,  $Y = e(g_p, g_p)^\omega$ ,  $\bar{Y} = e(g_p, g_p)^{\bar{\omega}}$ . The public key  $PK$  and master secret key  $MSK$  are defined as formula(1) and (2):

$$PK = \langle A_0, g_r, \{A_{i,t_i}\}_{1 \leq t_i \leq n_i, 1 \leq i \leq n}, Y, \bar{Y} \rangle \quad (1)$$

$$MSK = \langle g_p, \{a_{i,t_i}\}_{1 \leq t_i \leq n_i, 1 \leq i \leq n}, \omega, \bar{\omega} \rangle \quad (2)$$

- KeyGen(MSK, PK, L): For  $\forall i \in \mathbb{Z}_k$ , the DO randomly selects  $d_i \in \mathbb{Z}_N$ . Therefore, the  $SK_L$  is like the format of formula(3):

$$SK_L = \langle D_0 = g_p^{\omega - \sum_{i=1}^k d_i}, \bar{D}_0 = g_p^{\bar{\omega} - \sum_{i=1}^k d_i}, \{D_j = g_p^{\frac{d_i}{a_{i,t_i}}}\}_{1 \leq i \leq k} \rangle \quad (3)$$

- Encrypt(PK, m, T):

- a). DO selects random values  $R'_0 \in \mathbb{G}_r$  and  $s \in \mathbb{Z}_N$  to encrypt the message  $m \in \mathbb{G}$  as follows:

$$C_0 = A_0^s R'_0, \tilde{C} = m Y^s \quad (4)$$

- b). Let secret  $s$  be assigned in access tree and the root node of  $\mathcal{T}$  is set to be  $s$ . All child nodes except for leaf nodes are marked un-assigned status and root node is assigned status. Then, for each unassigned node, do the following operations recursively:

- If the node logic operator is *AND*, the first  $u - 1$  child nodes are set to  $s_i$  where  $s_i$  is randomly generated in  $\mathbb{Z}_N$ , and the last child node is set to  $s_u = s - \sum_{i=1}^{u-1} s_i$ ;
- If the node logic operator is *OR*, the child nodes of this node are set  $s$ ;
- If the node logic operator is *Threshold* with the threshold value  $h$ , it will arbitrarily generate a polynomial  $f$  that satisfies  $f(0) = s$  and the degree of polynomial is  $h - 1$ . For the  $i$ -th node, the value is assigned  $f(i)$ .

- c). Suppose that a *LPNode* in access tree has a shared secret  $s_\alpha$ . Then calculate a ciphertext component for each attribute in system according to formula(5), where  $s_{i,t_i} \in \mathbb{Z}_N$  is a random value and  $R'_{i,t_i}$  is also a random parameter produced from  $\mathbb{G}_r$ . Therefore, the ciphertext of *LPNode* is like formula(6):

$$C_{i,t_i} = \begin{cases} A_{i,t_i}^{s_{i,t_i}} R'_{i,t_i}, & L_i \in L_\alpha \text{ and } v_{i,t_i} \notin L_\alpha \\ A_{i,t_i}^{s_\alpha} R'_{i,t_i}, & \text{otherwise} \end{cases} \quad (5)$$

$$C_\alpha = \langle H_\alpha = \bar{Y}^{s_\alpha}, \bar{C}_\alpha = A_0^{s_\alpha} R_\alpha, \{C_{i,t_i}\}_{i \leq t_i \leq n_i, 1 \leq i \leq n} \rangle \quad (6)$$

where  $R_\alpha \in \mathbb{G}_r$ . Ultimately, output the final ciphertext:

$$CT = \langle \tilde{C}, C_0, \{C_\alpha\}_{\forall \alpha \text{ in } T} \rangle \quad (7)$$

- GenToken(CT,SK): DO selects a random factor  $z \in \mathbb{Z}_N$  and does the following operations:  $D_i^* = D_i^z, \bar{D}_0^* = \bar{D}_0^z, D_0^* = D_0^z, H_\alpha^* = H_\alpha^z$ . Finally, the DO output the TK as the formula(8) and CT to outsourced server by security channel.

$$TK = \langle D_i^*, \bar{D}_0^*, H_\alpha^* \rangle \quad (8)$$

- PreDecrypt(CT,TK,PK): In this process, outsourced server runs the following defined functions. If the node is a *LPNode*, it will execute  $PreDecNode(\alpha)$ . Similarly, if the node is a normal node, it will execute  $PreDecNode(\beta)$  where  $u$  is the number of child of node  $\beta$  and  $h$  is the threshold value of node  $\beta$ .

$$PreDecNode(\alpha) = \prod_{i=1}^k e(C_{i,t_i}, D_i^*) \quad (9)$$

PreDecNode( $\beta$ )

$$= \begin{cases} \prod_{i=1}^h PreDecNode(Child(\beta, i))^{\Delta_{i,\beta}(0)}, & Option(\beta) = Threshold \\ \prod_{i=1}^u PreDecNode(Child(\beta, i)), & Option(\beta) = AND \\ PreDecNode(Child(\beta, i)), & Option(\beta) = OR \end{cases} \quad (10)$$

- Testing: The outsourced server calculates  $\theta_\alpha = \frac{H_\alpha^*}{e(\bar{C}_\alpha, \bar{D}_0^*) PreDecNode(\alpha)}$ . If the *LPNode* satisfies the attributes represented by each leaf node under it, then it can obtain  $PreDecNode(\alpha) = e(g_p, g_p)^{zds_\alpha}$ . Further, the outsourced server performs the following calculations:

$$\begin{aligned} \theta_\alpha &= \frac{H_\alpha^*}{e(\bar{C}_\alpha, \bar{D}_0^*) PreDecNode(\alpha)} \\ &= \frac{\bar{Y}^{zs_\alpha}}{e(A_0^{s_\alpha} R_\alpha, g_p^{z(\bar{\omega}-d)}) e(g_p, g_p)^{zds_\alpha}} \\ &= \frac{e(g_p, g_p)^{\bar{\omega}zs_\alpha}}{e(g_p, g_p)^{zs_\alpha(\bar{\omega}-d)} e(g_p, g_p)^{zds_\alpha}} \\ &= 1 \end{aligned} \quad (11)$$

Evidently  $\theta_\alpha$  will be a random value, if the *LPNode* cannot satisfy attributes. When  $\theta_\alpha$  is a random value, it indicates this *LPNode* is an error node. Hence, there is no necessary to participate in the subsequent decryption phase. The next description operation is completed by executing formula(12) when  $\theta_\alpha = 1$ :

$$\begin{aligned} CT' &= \frac{1}{e(C_0, D_0^*) PreDecNode(root(T))} \\ &= \frac{1}{e(g_p^s R_0^s R_0', g_p^{z(\omega-d)}) e(g_p, g_p)^{dsz}} \\ &= \frac{1}{e(g_p, g_p)^{sz(\omega-d)} e(g_p, g_p)^{dsz}} \\ &= \frac{1}{e(g_p, g_p)^{sz\omega}} \end{aligned} \quad (12)$$

At last, outsourced server transfers  $CT'$  to DataUser.

- Decrypt( $CT', SK, CT$ ): DataUser receives  $CT'$  sent by outsourced server and use  $SK$  to decrypt ciphertext. The decryption process is described as formula(13):

$$\frac{\tilde{C}}{(CT')^{\frac{1}{z}}} = \frac{mY^s}{e(g_p \cdot g_p)^{s\omega}} = \frac{me(g_p \cdot g_p)^{s\omega}}{e(g_p \cdot g_p)^{s\omega}} = m \quad (13)$$

#### 4 Security Proof

In this section, we will give a proof of security of our scheme. The idea of Lai et.al inspires our thinking and based on dual encryption system invented by Waters et.al we completed our proof. Above all, for convenience, we denote the corresponding definitions used in Water's paper. Both ciphertext and private keys can take on one of two indistinguishable forms the *normal* or the *semi-functional* in dual system encryption system. A *normal* private key or ciphertext are generated from key generation or encryption algorithm. As for semi-functional private key, it can decrypt all normally generated ciphertext. However, decryption will fail if one try to decrypt a semi-functional ciphertext with a semi-functional privacy key. Similarly, semi-functional ciphertext can be decrypted only by normal private key. Note that the semi-functional ciphertext and semi-functional private keys are used in our proof process, and you can't search for them in the actual system.

The security of our scheme depends on three hypotheses which are mentioned in section 2.3 and we utilize hybrid argumentation techniques to prove the security of the scheme with the indistinguishability of a series of games. The followings displayed are the components of our proof procedure.

- Normal ciphertext: The normal ciphertext is like:

$$\langle \tilde{C}', C'_0, \{(C'_{i,t_i})_{1 \leq t_i \leq n_i, 1 \leq i \leq n}\} \rangle \quad (14)$$

- Semi-functional ciphertext: Select a random number  $x_0 \in \mathbb{Z}_N/0$  and in each  $LINode$ , for  $\forall t_i (i \in \mathbb{Z}_n/0, t_i \in \mathbb{Z}_{n_i}/0)$ , randomly generates  $x_{i,t_i} \in \mathbb{Z}_N/0$ , then, the semi-functional ciphertext is:

$$\langle \tilde{C} = \tilde{C}', C_0 = C'_0 g_q^{x_0}, \{C_{i,t_i} = C'_{i,t_i} g_q^{x_{i,t_i}}\}_{1 \leq t_i \leq n_i, 1 \leq i \leq n} \rangle \quad (15)$$

- $Game_{real}$ : The real game is composed of real private key and ciphertext;
- $Game_0$ : The query private key is normal but challenge ciphertext is semi-functional in  $Game_0$ ;
- $Game_{1,k}$ : It denotes the challenge ciphertext is semi-functional. The private key of the first queries is semi-functional and the remains are normal.
- $Game_{final}$ : The private key of all queries is semi-functional. The challenge ciphertext is a semi-functional ciphertext generated by encrypting a random message.

**Lemma 1** Suppose there exist a probabilistic polynomial time algorithm  $\mathcal{A}$  that can win  $\text{Game}_{\text{real}}, \text{Game}_0$  with advantage  $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{real}}} - \text{Adv}_{\mathcal{A}}^{\text{Game}_0} = \varepsilon$ . Then we can build a PPT algorithm  $\mathcal{C}$  to break Assumption1 with advantage  $\varepsilon$ .

*Proof* Given the distribution in Assumption1,  $D = \{\mathbb{G}, g_p, g_r\}$ .

- Setup: This phase is the same as Setup algorithm in section 3.3.
- Phase1: Adversary can answer private key of any attribute set.
- Challenge: Adversary selects two messages  $m_0, m_1$  with the same length and two access structures  $T_0, T_1$ . Note that any query cannot satisfy access structures  $T_0, T_1$ . Then adversary transfers them to challenger. After the challenger receives the parameters from adversary, challenger flips a coin  $b \in \{0, 1\}$  and encrypts the received message to  $m_b$  by using a access structure  $T_b$ . Afterwards, challenger randomly selects some necessary parameters,  $s \in \mathbb{Z}_N, R'_0 \in \mathbb{G}_r$  and generates  $\tilde{C}^* = m_b e(g_p, X)^{\omega s}, C_0^* = X^s R'_0$  which are components of ciphertext. When  $((L_i \in L_\alpha) \wedge (v_{i,t_i} \notin L_\alpha))$ , challenger obtains the random value  $s_{i,t_i} \in \mathbb{Z}_N$  and  $s_{i,t_i} \neq s_\alpha$ , set  $C_{i,t_i}^* = X^{a_{i,t_i} \cdot s_{i,t_i}} R'_{i,t_i}$ , otherwise, let  $C_{i,t_i}^* = X^{a_{i,t_i} \cdot s_\alpha} R'_{i,t_i}$ . Finally, challenger send the completed ciphertext  $\langle \tilde{C}^*, C_0^*, \{\{C_{i,t_i}^*\}_{1 \leq i \leq n, 1 \leq t_i \leq n_i}\}_{\forall \alpha} \rangle$  to  $\mathcal{A}$ .
- Phase2: It is the same as Phase1.
- Guess: Adversary outputs a guess  $d'$  to challenger.

Apparently, if  $X \in \mathbb{G}_p$  the answer ciphertext  $c^*$  must be normal ciphertext. Similarly, if  $X \in \mathbb{G}_{pq}$ , the answer ciphertext  $c^*$  have semi-function. Therefore, under the assumption that  $\mathcal{A}$  have a non-negligible advantage  $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{real}}} - \text{Adv}_{\mathcal{A}}^{\text{Game}_0} = \varepsilon$ , then challenger also can distinguish elements of  $\mathbb{G}_p$  or  $\mathbb{G}_{pq}$  with indistinguishable advantage.

**Lemma 2** If there exist a algorithm  $\mathcal{A}$  that can distinguish between  $\text{Game}_{\text{real}}$  and  $\text{Game}_0$  with advantage  $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{real}}} - \text{Adv}_{\mathcal{A}}^{\text{Game}_0} = \varepsilon$ , then a algorithm  $\mathcal{C}$  that can break Assumption2 with identical advantage  $\varepsilon$  in probabilistic polynomial time is constructed.

*Proof* Given the same conditions as Assumption2

- Setup: the similar step as in Lemma1.
- Challenge: In this phase,  $\mathcal{A}$  sends the followings that consists of  $m_0, m_1$  with the same bit-length and two access structure  $T_0, T_1$  (note that any attribute set can not satisfy them) to challenger. Subsequently, challenger selects a random value  $b \in 0, 1$  and use the access structure  $T_b$  to encrypt message to generate new ciphertext  $m_b$ . Then challenger arbitrarily chooses  $s \in \mathbb{Z}_N, R'_0 \in \mathbb{Z}_r$  and produces the semi-functional ciphertext by the corresponding parameters in  $MSK$  and assumption conditions  $\tilde{C}^* = m_b e(g_p, X_1 X_2)^{\omega s}, C_0 = (X_1, X_2)^s R'_0$ . Finally, when  $((L_i \in L_\alpha) \wedge (v_{i,t_i} \notin L_\alpha))$ , let  $C_{i,t_i}^* = (X_1 X_2)^{a_{i,t_i} s_{i,t_i}} R'_{i,t_i}$  where  $s_{i,t_i}$  is a random value and  $s_{i,t_i} \in \mathbb{Z}_N$  and  $s_{i,t_i} \neq s_\alpha$ , otherwise, let  $C_{i,t_i}^* = (X_1 X_2)^{a_{i,t_i} s_\alpha} R'_{i,t_i}$ . Hence, the ciphertext  $\langle \tilde{C}^*, C_0^*, \{\{C_{i,t_i}^*\}_{1 \leq i \leq n, 1 \leq t_i \leq n_i}\}_{\forall \alpha} \rangle$  is transmit to  $\mathcal{A}$ .

- Phase1 and Phase2: Considering the  $i$ -th query of  $\mathcal{A}$ , obviously, the private key query is divided the following three sections:
  - Case1: When  $i > k$ , challenger can correctly calculate normal private key using  $MSK$  and attribute sets from the query of  $\mathcal{A}$ , then challenger sends the normal private key to  $\mathcal{A}$ .
  - Case2: When  $i < k$ , according to the above definition of  $Game_{1,k}$ , the first  $k-1$  private key of  $Game_{1,k-1}$  and  $Game_{1,k}$  is semi-functional. Therefore, challenger calculates to the semi-functional key and sends it to  $\mathcal{A}$ .
  - Case3: When  $i = k$ , for  $\forall i \in \mathbb{Z}_k$ , challenger randomly chooses  $d_i \in \mathbb{Z}_N$  and generates private key  $D'_0 = X^{\omega-d}$ ,  $D'_i = X^{\frac{d_i}{a_{i,t_i}}}$  where  $d_i \in \mathbb{Z}_N, d_i$  is a random value and sets  $d = \sum_{i=1}^k d_i$ .
- Guess: Adversary outputs a guess  $d'$  to challenger.

One situation is that if  $X \in \mathbb{G}_p$ , let  $X = \mathbb{G}_p$ , hence, the generated key is normal private key, further, the game is  $Game_{1,k-1}$ . The other is that  $X = g_p g_q^{\frac{y_0}{\omega-d}}$ ,  $X \in \mathbb{G}_{pq}$ , moreover, the generated private key is semi-functional key and the game is  $Game_{1,k}$ . Therefore, considering the above two situation, if  $\mathcal{A}$  has a non-negligible advantage  $Adv_{\mathcal{A}}^{Game_{1,k-1}} - Adv_{\mathcal{A}}^{Game_{1,k}} = \varepsilon$ , then challenger also has a non-negligible advantage to distinguish the elements between  $\mathbb{G}_{pq}$  and  $\mathbb{G}_p$ .

**Lemma 3** *If there exist a probabilistic polynomial time algorithm  $\mathcal{A}$  that has the advantage  $Adv_{\mathcal{A}}^{Game_{1,u}} - Adv_{\mathcal{A}}^{Game_{final}} = \varepsilon$ , then a probabilistic polynomial time algorithm  $\mathcal{C}$  that is able to break Assumption3 with the advantage  $\varepsilon$  can be built.*

*Proof* Given the distributions in Assumption3,  $D = (\mathbb{G}, g_p, g_p^\omega X_2, g_p^s Y_2, Z_1 Z_2 g_r)$ .

- Setup: Repeat the step in Lemma1.
- Phase1 and Phase 2: In  $Game_{1,u}$  and  $Game_{final}$ , the generated private keys are semi-functional. The challenger produces semi-functional private keys  $D'_0 = g_p^\omega X_2 (Z_1 Z_2)^{-d}$ ,  $D'_i = (Z_1 Z_2)^{\frac{d_i}{a_{i,t_i}}}$  based on queried attributes sets by  $\mathcal{A}$  where  $i \in \mathbb{Z}_k, d_i \in \mathbb{Z}_N$  are random values and set  $d = \sum_{i=1}^k d_i$ .
- Challenge:  $\mathcal{A}$  arbitrarily selects two messages  $m_0, m_1$  with the same length and two access structures  $T_0, T_1$  that cannot be satisfied by any attribute set. Then  $\mathcal{A}$  sends them to challenger. Challenger use access structure  $T_b$  to encrypt the message to generate  $m_b$  or a random value  $r$ . Subsequently, challenger arbitrarily chooses  $s$  and do the following operations to produce component of ciphertext.  $\tilde{C}^* = m_b T, C_0^* = g_p^s Y_2 R'_0$ . The remain ciphertext components are obtained by the following calculations. When  $(L_i \in L_\alpha) \wedge (v_{i,t_i} \notin L_\alpha)$ , let  $C_{i,t_i}^* = g_p^{a_{i,t_i} s_{i,t_i}} R'_{i,t_i}$  where  $s_{i,t_i} \in \mathbb{Z}_N$  and  $s_{i,t_i} \neq s_\alpha$ , otherwise,  $C_{i,t_i}^* = g_p^{a_{i,t_i} s_\alpha} R'_{i,t_i}$ . Lastly, the whole ciphertext  $\langle \tilde{C}^*, C_0^*, C_{i,t_i}^* \rangle$  is dispatched to  $\mathcal{A}$ .
- Guess: Challenger outputs a guess marked  $d'$  of  $d$ .

When  $T = e(g_p, g_p)^{\omega s}$ , encrypted message  $m_b$  is a semi-functional ciphertext, otherwise, let  $T = e(g_p, g_p)^{\omega s - x_0}$ , the generated semi-functional ciphertext belongs to the random message  $m_b e(g_p, g_p)^{\omega s - x_0}$ . Therefore, If  $\mathcal{A}$  has a non-negligible advantage  $Adv_{\mathcal{A}}^{Game_{1,u}} - Adv_{\mathcal{A}}^{Game_{final}} = \varepsilon$ , then challenger can also distinguish a random element between  $e(g_p, g_p)^{\omega s}$  and  $\mathbb{G}_T$  with non-negligible advantage.

**Definition 5** Assume that Assumption1, Assumption2 and Assumption3 hold, then the scheme described in this paper is secure.

*Proof* If Assumption1, Assumption2 and Assumption3 hold,  $Game_{real}$  and  $Game_{final}$  are indistinguishable, furthermore,  $\mathcal{A}$  has a negligible advantage  $Adv_{\mathcal{A}}^{Game_{final}}$  in  $Game_{final}$ , hence,  $\mathcal{A}$  have a negligible advantage  $Adv_{\mathcal{A}}^{Game_{real}}$  in  $Game_{real}$ . So, our scheme satisfies security under Chosen Ciphertext Attack(CCA).

## 5 Performance Analysis

In this section, we reveal the performance of our scheme and it is divided into two sections theoretical analysis and experimental simulation. In the former, we mainly adopt mathematical methods to analyze the efficiency of our scheme and the length of ciphertext, and compare it with other schemes. In order to verify the correctness of the theoretical analysis scheme, we conducted a simulation experiment in the latter.

### 5.1 Theoretical Analysis

Table1 indicates the properties between our scheme and other related scheme.  $\checkmark$  means the scheme satisfies this property, conversely,  $\times$  shows the schemes can not match this property. From Table1, we can get the conclusion that compared with other schemes, our scheme not only has strong expressive ability, but also has more comprehensive features to improve efficient.

Firstly, let us present the notations appearing in the following.  $N$  expresses the number of attributes in the whole system.  $T_{\mathbb{G}}, T_{\mathbb{G}_T}$  indicate the cost time of operations of group  $\mathbb{G}$  and group  $\mathbb{G}_T$ . Note that the operations only include exponentiation for convenience.  $|*|$  is the bit-length of  $*$ .

**Table 1** PROPERTY COMPARISON TABLE

Scheme	Access Structure	Hidden Policy	Testing	Outsourcing	Security Model
Nishide	AND-gates	$\checkmark$	$\times$	$\times$	selectively-CCA
Lai	AND-gates	$\checkmark$	$\times$	$\times$	adaptively-CCA
Luan	Tree	$\times$	$\times$	$\times$	selectively-CPA
Li	AND-gates	$\checkmark$	$\checkmark$	$\times$	selective-CPA
Xu	Tree	$\checkmark$	$\times$	$\times$	selectively-CPA
Ours	Tree	$\checkmark$	$\checkmark$	$\checkmark$	adaptively-CCA

---

## 5.2 Experimental Simulation