A Tree-based CP-ABE Scheme with Hidden Policy Supporting Secure Data Sharing in Cloud Computing

Runhua Xu[*], Yang Wang[†], Bo Lang[*]
[*]State Key Laboratory of Software Development Environment
School of Computer Science and Engineering, Beihang University
Beijing, China
xurunhua@nlsde.buaa.edu.cn, langbo@buaa.edu.cn
[†]National Computer Network Emergency Response Technical Team/Coordination Center of China
Beijing, China
aaron@ncic.ac.cn

***Abstract*—With flexible and scalable features for fine-grained access control, Ciphertext Policy Attribute-based Encryption (CP-ABE) is widely used as a kind of data protection mechanism in the cloud computing. However, the access policy of CP-ABE scheme may contain sensitive information which causes privacy revelation of the data provider or receiver. Some papers proposed hidden policy CP-ABE scheme, which are based on And-gate access structure whose expressive ability of access policy is limited. CP-ABE with the tree-based access structure has stronger expressive ability and more flexible access control capability. Therefore, it has broad application prospects compared to other mechanisms. This paper proposes a tree-based access structure CP-ABE scheme with hidden policy (CP-ABE-HP), which can both protect the policy and has flexible access control capability. We prove the Chosen-plaintext Attack (CPA) security of our scheme under the Decisional Bilinear Diffie-Hellman (DBDH) assumption in the standard model.**

Keywords-ciphertext policy attribute-based encryption; access control; hidden policy; cloud computing

I. INTRODUCTION

With the development of cloud computing, the cloud storage is becoming a popular way of data storage for enterprises or individuals. In the cloud storage environment, the protective ability of data becomes very important, because data is out of the user's control domain and the cloud storage service provider may be unreliable. Data encryption, currently the primary means to protect data, cannot satisfy data protection requirements in various online applications which own a large amount of users, due to its complex key management mechanism and poor scalability. Therefore, a new data oriented protection mechanism is urgently needed, in which data has the ability to protect its confidentiality and integrity all by itself rather than depending on the cloud storage server. We call this kind of new data protection the data-centric self-contained protection. Also we believe that this kind of data protection can be achieved by integrating data encryption with data access control.

In recent years, for the situation of uncertain decrypting party, Attribute-based Encryption (ABE)[1] mechanism, proposed by Sahai and Waters, was developed based on Identity-based encryption mechanism. Without knowing the specific decrypting parties, the data provider encrypts data according to an access structure consisting of a series of attribute descriptions, and the data receiver can decrypt the ciphertext only if he/she satisfies the attribute descriptions in the access structure. With data decryption depending on user attributes, ABE can solve the security problems of outsourced data effectively. At present, ABE can be divided into two types: Key Policy Attribute-based Encryption (KP-ABE) and Ciphertext Policy Attribute-based Encryption (CP-ABE). In KP-ABE, which was proposed by Goyal et al [2], the ciphertext is associated with a set of attributes and the secret key is associated with the access structure. On the contrary, in CP-ABE, proposed by Bethencourt et al [3], the ciphertext is associated with the access structure and the secret key is associated with a set of attributes. In both cases, only when the user's attributes satisfy the access policy related to the ciphertext, can he/she decrypt the ciphertext successfully. Consequently, the CP-ABE scheme is more suitable for data-centric self-contained protection in cloud storage environment. The CP-ABE scheme currently has three kinds of access structures: And-gate access structure, tree-based access structure and Linear Secret Share Scheme (LSSS) matrix access structure. The tree-based access structure can express much more complex access policy with its hierarchy. Thus, it has a more flexible access control capability for data encryption. Therefore, CP-ABE has gained much more attentions in cloud computing.

In the originally proposed CP-ABE scheme, the access structure is embedded in the ciphertext and whoever obtains the ciphertext can see the content of the access structure. However, this full exposure of data's access policy will disclose sensitive information of the decryption or encryption party. For example, in commercial environments, the access structure may contain trade secrets; an access structure for secret job information released by a company on public platform may disclose the company's development direction and strategy. In military applications, the access structures themself may contain military secrets, such as sensitive information like organizational structure, the core combat troops, staff ratio, and firepower configuration of a group army. When utilizing ABE to protect shared data on the Internet, the access policy may also disclose the receivers' privacy information. Meanwhile, in order to avoid

the attack by malicious users and policy-based inference for important information, the policy should be hidden.

The anonymous ABE schemes [4-6] give a good solution to these problems. These schemes protect users' privacy information by establishing security protocols and using encryption to protect access policy against unauthorized access in the security protocols. Recently, the researchers also proposed a series of CP-ABE schemes [7-9] with hidden policy. But these schemes are based on simple And-gate access structure and their policy expressive ability is limited. To the best of our knowledge, the tree-based access structure CP-ABE scheme with hidden policy has not been proposed at present.

This paper proposes a tree-based access structure CP-ABE scheme with hidden policy (CP-ABE-HP) which is proved to have the Chosen-plaintext Attack (CPA) security under the Decisional Bilinear Diffie-Hellman (DBDH) assumption in the standard model. Inspired by the tree-based access structure in ITHJ09 scheme [10] and the policy hiding method in And-gate access structure CP-ABE which proposed by Xiaohui et al [7], we proposed a more efficient and stronger expressive CP-ABE scheme with hidden policy. Our scheme also uses subgroup element's orthogonal property in composite order bilinear groups and introduces some random elements into the policy key component.

The remaining sections are organized as follows. In Section 2, we introduce related work. In Section 3, we review some preliminary concepts. We propose the CP-ABE-HP scheme and give analysis in Section 4. In Section 5 we give the security proof of our scheme. We then give an implementation framework of our scheme in cloud computing environment in Section 6. Finally, we conclude the paper in Section 7.

II. RELATED WORK

BSW07 [3] scheme, proposed by Bethencourt et al, uses tree-based access structure to express access policy, which supports AND, OR and of threshold operator. Its ciphertext length, time of encryption and decryption, and the number of attributes in the access structure are linear correlation. But the security proof is based on common group model, rather than the standard numerical theoretical assumptions. Cheung and Newport [11] primarily constructed the CPA security CP-ABE mechanism (CN07) based on the DBDH assumption. However, its access structure only supports AND, OR operation with weak expressive ability. Also, the ciphertext and key length are linear with the number of system attributes and scheme efficiency is lower.

Goyal et al proposed the Bounded Ciphertext Policy Attribute-based Encryption (BCP-ABE)[12] with tree-based access structure supporting *AND*, *OR* and *of* threshold operator, but the height of the tree and number of child non-leaf nodes are limited. Ibraimi et al [10] gave a CP-ABE mechanism (ITHJ09) based on DBDH assumption using Shamir secret sharing technology [13] to support AND, OR and of threshold operation. Its access structure is an n-ary tree and its key generation and decryption or encryption cost is lower than BSW07 scheme. Waters [14] firstly used the Linear Secret Sharing Scheme matrix to express access policy.

Kapadia et al [15] gave a scheme with hidden certificate and hidden policy based on PEAPOD system. The scheme introduced an online semi-trusted server, but cannot prevent collusion attacks. To prevent users' collusion attacks, Yu et al constructed two kind of anonymous CP-ABE mechanism [5, 16] used in CDN network and multicast user groups. However, these anonymous mechanisms used strong security assumption, so the security level is lower. Nishide et al [6] firstly proposed anonymous CP-ABE with hidden policy based on DBDH assumption and D-Linear assumption. But the mechanism only supports AND-gate access structure. In papers [8, 17-19], the authors have proposed different ways to deal with policy hiding issues. Lai et al [8] used inner product Predicate Encryption technology to achieve the hidden policy CP-ABE scheme in fully security model. After that, they also proposed a partial hidden policy scheme [20] based on LSSS matrix access structure and pointed out that the structure is more flexible than other scheme [6, 8, 21]. Among these schemes, the privacy-aware ABE proposed by Jin et al [21] aimed at the prevention of users' collusion attacks. And their main idea was binding the user's ID to detect whether a user shares their property keys or not. Balu et al [17, 19] calculated dual key for each attribute element to achieve anonymous policy or privacy preserving without supporting *of* threshold operator. With And-gate access structure supporting negative attribute and wildcard, a hidden policy scheme, proposed by Nishanth and Devesh [9], focused on the constant size of ciphertext and key. Xiaohui et al also proposed a hidden policy scheme which used And-gate access structure with provably security under the standard model, and it based on Waters' scheme [22].

Considering the security and expressive ability of access policy, only the W08 and ITHJ09 scheme support the AND, OR and threshold operations under the standard numerical theoretical assumptions, and the computation cost of encryption and decryption of ITHJ09 is lower than W08's. Meanwhile, in terms of hidden policy, all existing CP-ABE schemes are based on And-gate access structure. Though their efficiency has been improved, the expressive ability of policy is limited. In the background of cloud storage applications, CP-ABE scheme with flexible policy expression ability will have broad application prospects. Therefore, the paper focuses on the research of tree-based access structure CP-ABE scheme with hidden policy.

III. PRELIMINARIES

A. Composite order bilinear groups

Composite order bilinear groups were first introduced by Boneh et al [23]. The order of bilinear groups we used is the product of two distinct primes. Let p, r be distinct primes, G and G_T be cyclic groups of order $N=pr$. And $e:G\times G\rightarrow G_T$ is a map satisfied the following conditions:

- Bilinear: $\forall\ g, h\in G, a, b\in Z_N, e(g^a, h^b) = e(g, h)^{ab}$.
- Non-degenerate: $\exists\ g\in G$ such that $e(g, h)$ has order N in G_T.

We use G_p and G_r to denote the subgroups of G with order p and r respectively. Note also that if $h_p \in G_p$ and $h_r \in G_r$ then $e(h_r, h_p) = 1$.

B. The Decisional Bilinear Diffie-Hellman Assumption

In this paper, we use DBDH assumption as the complexity assumption. Let $e: G \times G \to G_T$ be an efficiently computable bilinear map and g is the generator of G. Choose random numbers $a,b,c,z \in Z_p$. The DBDH assumption is that no probabilistic polynomial-time algorithm β can distinguish the tuple $(g, g^a, g^b, g^c, e(g, g)^{abc})$ from the tuple $(g, g^a, g^b, g^c, e(g, g)^z)$ with more than a negligible advantage.

C. Access Structure

Definition 1: Access Structure [24]: Let $\{p_1, p_2, \ldots, p_n\}$ be a set of parties. A collection $A \in 2^{\{p1, p2, \ldots, pn\}}$ is monotone if $\forall B,C$: if $B \in A \wedge B \subseteq C$ then $C \in A$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) A of non-empty subsets $\{p_1, p_2, \ldots, p_n\}$, i.e., $A \in 2^{\{p1, p2, \ldots, pn\}} \setminus \{\varnothing\}$. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.

In CP-ABE-HP mechanism, we use attributes instead of the p_i and the access structure A will contain the set of authorized attributes.

D. CP-ABE

The ciphertext-policy attribute based encryption(CP-ABE) scheme consists of four fundamental algorithms [3]: Setup, Encrypt, Key Generation, and Decrypt.

- 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.
- 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.
- 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.
- 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.

E. Security Model

The CPA semantic security model of CP-ABE-HP will be the based on the IND-sAtt-CPA game [10], which is a simulation between a challenger and an adversary A. In the game, the challenger simulates an execution environment of algorithms to answer the adversary's query request. The specific game process is as follows:

- Init Phase. The adversary chooses a challenge access tree and gives it to the challenger.
- Setup Phase. The challenger runs Setup algorithm to generate (PK, MK) and gives the public key PK to adversary A.
- Phase 1. Adversary A makes a secret key request to the key generation oracle for any attribute sets. The challenger runs Key-Generation (MK, S) algorithm to generate a private key.
- Challenge Phase. Adversary A sends to the challenger two equal length messages m_0, m_1. The challenger picks a random bit $b \in \{0, 1\}$ and returns c_b = Encrypt (m_b, τ^*, PK).
- Phase 2. Adversary A can continue querying key generation oracle with the same restriction as in Phase 1.
- Guess Phase. Adversary A outputs a guess $b' \in \{0, 1\}$.

Definition 2: if the attack advantage of adversary is ignored in the IND-sAtt-CPA game in any polynomial time, the CP-ABE-HP scheme can be at CPA security. And the advantage is $\varepsilon = |\Pr - 1/2|$.

IV. CP-ABE-HP

A. CP-ABE-HP Scheme

The specific CP-ABE-HP scheme is as follows:

1) Setup (k): the algorithm takes security parameter k as input and generates the following parameters.

a) Generate the bilinear groups G and a bilinear map $e: G \times G \to G_T$, and G and G_T are the cyclic groups of order $N=pr$, where the p and r are distinct primes. Let G_p and G_r be the subgroup of the G with order p and r respectively. Also g_p and g_r are the generator of G_p and G_r respectively.

b) Generate the attribute set $U=\{a_1, a_2, \ldots, a_n\}$, random element α, $t_1, t_2, \ldots, t_n \in Z^*_p$ and $R_0, R_1, R_2, \ldots, R_n \in G_r$. Calculate the public key as follows:

$$x = g_p \cdot R_0 \tag{1}$$

$$y = e(g_p, g_p)^{\alpha} \tag{2}$$

$$T_j = g_p^{t_j} \cdot R_j (1 \le j \le n) \tag{3}$$

So, the public key is $pk = (e, x, y, T_j(1 \le j \le n))$, and the master key is $mk = (\alpha, t_j(1 \le j \le n))$.

2) Key-Generation (w, mk): the algorithm takes w and mk as the input, where w is the attribute set submited by the user and mk is the master key. The detail algorithm is as follows:

a) Choose a random element $r \in Z^*_p$, calculate:

$$d_0 = g_p^{\alpha - r} \tag{4}$$

b) For every attribute a_j in w, calculate:

$$d_j = g_p^{rt_j^{-1}} \tag{5}$$

Return the secret key $sk_w = (d_0, \forall\ a_j \in w : d_j)$.

3) Encrypt (m, τ, pk): the algorithm encrypts a message $m \in G_T$ as follows, where m is the message, τ is the access policy tree and pk is the public key of the system.

a) Select a random element $s \in Z^*_p$, $R'_0 \in G_r$, calculate:

$$c_0 = x^s \cdot R_0' \tag{6}$$

$$c_1 = m \cdot y^s = m \cdot e(g_p, g_p)^{\alpha s} \tag{7}$$

b) Assign the secret s in the tree-based access policy: set the value of the root node of τ to be s. Make all child nodes as un-assigned and the root node as assigned. Recursively, for each un-assigned non-leaf node, do as follows:

- If the node operator is *of* (threshold operator) and its child nodes are un-assigned, the secret s is divided by (t,n)-Shamir Secret Sharing, where n is the number of all child nodes and t is number of child nodes for recover secret s. For each child node, its sharing secret value is $s_i = f(i)$ and mark this node as assigned.
- If the node operator is $\wedge$ and its child nodes are un-assigned, ibid, using (t,n)-Shamir Secret Sharing to share the secret s, where t=n. For each child node, its sharing secret value is $s_i = f(i)$ and mark this node as assigned。
- If the node operator is $\vee$ and its child nodes are un-assigned, ibid, using (t,n)-Shamir Secret Sharing to share the secret s, where t=1. For each child node, its sharing secret value is $s_i = f(i)$ and make this node as assigned.

Note that i denotes the position index of the leaf node and the value of each leaf node is used to generate the ciphertext component. The function $f(x)$ is a random polynomial over Z^*_p, and defined as follows:

$$f(x) = \sum_{j=0}^{t-1} b_j x^j \tag{8}$$

where b_j is a random coefficient and t is the number of child nodes.

c) For each leaf node, calculate as follows:

$$\forall a_{j,i} \in \tau, c_{j,i} = T_j^{s_i} \cdot R_j' \tag{9}$$

Where i denotes the index of leaf node in the tree, and R'_j is a random element in G_r group.

Return the ciphertext $c_\tau = (c_0, c_1, \forall\ a_{j,i} \in \tau : [i, c_{j,i}])$.

4) Decrypt (c_τ, sk_w): the algorithm is described as follows:

$$m' = \frac{c_1}{e(c_0, d_0) \cdot \prod_{a_j \in w} e(c_{j,i}, d_j)^{l_i(0)}} \tag{10}$$

Where $l_i(0)$ is Lagrange coefficient, can be calculated by the attribute index i, which can be found in the ciphertext components of the attributes, namely, $[i, c_{j,i}]$. And the input parameters c_τ, sk_w denote the ciphertext, the users' private key respectively.

B. Analysis

Correctness Proof:

We give the correctness proof as follows:

$$\begin{aligned} m' &= \frac{c_1}{e(c_0, d_0) \cdot \prod_{a_j \in w} e(c_{j,i}, d_j)^{l_i(0)}} \\ &= \frac{m \cdot e(g_p, g_p)^{\alpha s}}{e(g_p^s, g_p^{\alpha - r}) \cdot e(R_0^s \cdot R_0', g_p^{\alpha - r})} \\ &\cdot \frac{1}{\prod_{a_j \in w} (e(g_p^{t_j s_i}, g_p^{rt_j^{-1}})^{l_i(0)} \cdot e(R_j^{s_i} \cdot R_j', g_p^{rt_j^{-1}})^{l_i(0)})} \\ &= \frac{m \cdot e(g_p, g_p)^{\alpha s}}{e(g_p^s, g_p^{\alpha - r}) \cdot e(g_p, g_p)^{\sum rs_i l_i(0)}} \\ &= \frac{m \cdot e(g_p, g_p)^{\alpha s}}{e(g_p^s, g_p^{\alpha - r}) \cdot e(g_p, g_p)^{rs}} \\ &= m \end{aligned} \tag{11}$$

In previous schemes [3,7-11], the access policy was appended to the ciphertext. Because the access policy was public, it's useless to protect the ciphertext components of the attributes that were associated with the tree-based access policy. However, when the policy was hidden, the only way to find some information about the access policy was to attack the ciphertext components of the attributes. Hence, it is necessary to protect these parts of the ciphertext. In our scheme, we use the property of composite order bilinear groups to achieve the

TABLE I. COMPARISON OF OUR SCHEME WITH OTHER SCHEMES IN COMPUTING COST

Scheme	Access Structure	Hidden Policy	Encrypt	Decrypt
CN07[11]	And-gate	N	$(n+1)G+2G_t$	$(n+1)C_e+(n+1)G_t$
Emura09[25]	And-gate	N	$(n+1)G+2G_t$	$2C_e+2G_t$
Xiao12[7]	And-gate	Y	$(n+3)G+2G_t$	$2C_e+2G_t$
BSW07[14]	Tree	N	$(2\|A_c\|+1)G+2G_t$	$2\|A_u\|C_e+(2\|S\|+2)G_t$
ITHJ09[10]	Tree	N	$(\|A_c\|+1)G+2G_t$	$(\|w\|+1)C_e+(\|w\|+1)G_t$
CP-ABE-HP	Tree	Y	$2(\|A_c\|+1)G+2G_t$	$(\|w\|+1)C_e+(\|w\|+1)G_t$

[a] Note: G and Gt represent the computing on G and Gt groups respectively. |w| is the number of user's attributes. Ce denotes the bilinear map computing. | Ac | stands for the attribute number in the access structure. | Au | is the leaf node number in the access structure. |S| indicates the number of user's attribute associated with the private key.

goal of anonymous attributes of receivers. The most important part is that the random element is introduced into ciphertext of c_0 and $c_{j,i}$. In encryption phase, c_0 and $c_{j,i}$ multiplies by the random elements R_0 and R' of G_r respectively, as shown in equation (6) and (9). Meanwhile, it does not affect the decryption result in the decryption phase, as shown in equation (11). So, it can effectively prevent some malicious attacker from testing the access policy by a possible access structure *w'*, guessing the access structure, and getting the anonymous information of receivers.

The performance analysis of the computing efficiency is shown in Table I. CN07, Emura09 and Xiao12 are the CP-ABE schemes based on And-gate access structure. Emura09 and Xiao12 scheme have the constant size of the ciphertext and the private key, and Xiao12 scheme realizes a hidden policy scheme. Compared to Emura09 scheme, the Xiao12 scheme has 2 additional computing costs in *G* group during encryption phase to achieve hidden policy, which is necessary for goal of policy hiding. BSW07, ITHJ09 and our scheme all use tree-based access structure, and our scheme increases one and (|Ac|+1) computing cost on *G* group compared with BSW07 and ITHJ09 respectively, however the computing was only the non-exponentiation on *G* group. Compared with And-gate hidden access policy in Xiao12 scheme, the computing consumption of our scheme is more than Xiao12 scheme during the decryption phase. The reason is that Xiao12 scheme has the constant size of ciphertext and private key. But during the encryption phase, the computing cost of our scheme is lower than Xiao12 scheme.

V. SECURITY PROOF

In this section, we give the security proof of CP-ABE-HP scheme. Firstly, we suppose that the IND-sAtt-CPA game can be won by an adversary *A* with a non-negligible advantage ε. We will build a simulator β, which has the ability to solve the DBDH assumption problem with advantage $\varepsilon/2$ from the attack ability of adversary *A*. The simulator firstly sets the bilinear group *G* of order $N=pr$ and the bilinear map $e: G\times G\rightarrow G_T$, where *p* and *r* are the distinct primes and *G* and G_T are cyclic groups. Let G_p and G_r be the subgroup of *G* with order *p* and *r* and generator g_p and g_r respectively. The challenger selects $u = {}_R\{0, 1\}$ and sets Z_u as follows:

$$\begin{cases} Z_u = e(g_p, g_p)^{\theta}, u = 0 \\ Z_u = e(g_p, g_p)^{abc}, u = 1 \end{cases} \quad (12)$$

And then the challenger sends a DBDH challenge $(g_p, A, B, C, Z_u) = (g_p, g_p{}^a, g_p{}^b, g_p{}^c, Z_u)$ to the simulator.

In the attack game, the simulator plays the challenger role of adversary and we refer to it as the challenger in the following IND-sAtt-CPA game:

- Init Phase. The adversary chooses a challenge access τ^* and sends it to the challenger.
- Setup Phase. The challenger selects a random element $x' \in Z_p$ and sets $\alpha = ab + x'$, then calculates:

$$y = e(g_p, g_p)^{\alpha} = e(g_p, g_p)^{ab} e(g_p, g_p)^{x'} \quad (13)$$

Select following elements randomly: $t_j \in {}_R Z^*_p$, R_j, $R_0 \in {}_R G_r, (1 \le j \le n)$, and calculate:

$$\forall a_j \in U : T_j = \begin{cases} g_p^{b/t_j} \cdot R_j, a_j \notin \tau^* \\ g_p^{t_j} \cdot R_j, a_j \in \tau^* \end{cases}, (1 \le j \le n) \quad (14)$$

After setting the parameters, the challenger sends the adversary *A* the following public key *pk* $(x=g_p \cdot R_0, y, T_j, (1\le j\le n))$.

- Phase 1. The adversary sends a user private key query request to the challenger by any attributes set. And the attributes set is as follows:

$$w_j = \{a_j \mid a_j \in \Omega\}, (a_j \notin \tau^*) \quad (15)$$

For each query request of the adversary, the challenger selects random element $r' \in {}_R Z_p$ and sets $r = ab + r'b$, so

$$d_0 = g_p^{\alpha-(ab+r'b)} = g_p^{x'-r'b} = g_p^{x'} (g_p^b)^{-r'} \quad (16)$$

As the restriction $a_j \notin \tau^*$ in the attributes set of private key request from the adversary, we have the following result:

$$d_j = g_p^{rt_j/b} = (g_p^a)^{t_j} g_p^{r't_j}, (a_j \notin \tau^*) \quad (17)$$

And the challenger sends the adversary the user private key: $sk_w(d_0, \forall\ a_j \in w_j : d_j)$.

- Challenge Phase. The adversary submits two plaintext messages m_0, m_1 to the challenger. And the challenger selects a random plaintext message m_b from the two messages, where $b \in_R \{0,1\}$. Encrypt the message as follows:

$$c_0 = g_p^c \cdot R_0^c \cdot R_0^{'} \tag{18}$$

$$c_1 = m_b e(g_p, g_p)^{abc} e(g_p^c, g_p^{x'}) \tag{19}$$

Then set the root node value of challenge tree τ^* to be g_p^c, and initialize all child nodes as un-assigned and mark the root node as assigned. Recursively, for each un-assigned non-leaf, if the node's child nodes are un-assigned, the challenger select a polynomial *f(i)*, *i* donating the attribute index of challenge tree and *f(0)=c*. For each child node, the challenger assigns a value $g^{f(i)}$, and marks this node as assigned. The polynomial *f(i)* is set with the following rule:

- If the node symbol is *of* (threshold operator), set the polynomial *f(i)* of degree *t*-1, where *t* denotes the number of nodes to recover the secret.
- If the node symbol is $\wedge$, set the polynomial *f(i)* of degree *n*-1, where n denotes the number of all leaf nodes.
- If the node symbol is $\vee$, set the polynomial *f(i)* of degree 0, so the *f(i)* is constant number and assign it to each of its child node.

- Phase 2. The adversary continues to send the secret key requests to the challenger with the same restriction as in Phase 1.
- Guess Phase. The adversary outputs a guess $b' \in \{0, 1\}$.

If $b' = b$, the challenger can guess that $u = 0$, $Z_u = e(g_p, g_p)^{abc}$. As $Z_u = e(g_p, g_p)^{abc}$ is a reasonable simulation of the simulator, the ciphertext is a valid ciphertext in the system. Hence, with the help of adversary, the challenger solves the DBDH assumption problem with the following advantage:

$$\Pr[b' = b \mid Z_u = e(g_p, g_p)^{abc}] = 1/2 + \varepsilon \tag{20}$$

Otherwise, the challenger guesses that $u = 0$, $Z_u = e(g_p, g_p)^{\theta}$. Right now the value of $Z_u = e(g_p, g_p)^{\theta}$ is a random ciphertext relative to the adversary. And the adversary cannot get any information about the plaintext message m_b. So, the challenger solves the DBDH assumption problem with the following advantage:

$$\Pr[b' \neq b \mid Z_u = e(g_p, g_p)^{\theta}] = 1/2 \tag{21}$$

Conclusion as a result, for any guesses, the challenger solves the DBDH assumption problem with the following advantage:

$$\frac{1}{2}\Pr[u' = u \mid u = 0] + \frac{1}{2}\Pr[u' = u \mid u = 1] - \frac{1}{2} = \frac{\varepsilon}{2} \tag{22}$$

In summary, the elements, like R_j, $R_0^{'}$, from the G_r group are random and one-time elements. Compared to the previous schemes [3,7-11], the random elements do not lead to new security problems. Hence, we focus on the same security model as before. If the adversary has the above advantage ε to win the IND-sAtt-CPA game, the challenger will solve the DBDH assumption problem with advantage $\varepsilon/2$ by the help of the adversary's advantage. However, there are no effective polynomial algorithms which can solve the DBDH assumption problem with non-negligible advantage according to the DBDH assumption. Hence, the adversary also cannot win the IND-sAtt-CPA game with the above advantage ε, namely, the adversary having no advantage to break through CP-ABE-HP system.

VI. An Implementation Framework of CP-ABE-HP

We have implemented a system for data sharing based on the CP-ABE-HP scheme. The system can be used to protect personal data stored in the private cloud or public cloud. The framework is shown in Fig.1.

The system needs to be initialized by distributing the public parameters to users. If a user wants to share his data with a group of specific users, he just needs to encrypt the data with the access policy under the encrypt algorithm of our scheme, and then he shares the data to the cloud. When the recipient gets the encrypted data from the cloud, he should firstly acquire the private key from the security server. The security server contains two parts, the Attribute Authority(AA) and the Private Key Generator(PKG). The AA firstly authenticates the recipient's attributes, and then asks the PKG to generate the private key containing these authenticated attributes for the recipient, and lastly the AA sends the private key to the recipient. If the recipient's attributes in his private key satisfy the access policy which is corresponding to the ciphertext, he can decrypt the data successfully. Otherwise, he can't decrypt it.

VII. Conclusions

In CP-ABE schemes, policy hidden is of great significance in certain applications for protecting the privacy information of data provider and receiver. By introducing random element of subgroups into the policy key components, and with the property of subgroup element's orthogonal in composite order bilinear groups, the paper proposed the CP-ABE-HP scheme, which effectively realizes policy hidden in encryption. Meanwhile the tree-based access structure of CP-ABE-HP ensures that users can define their policies flexibly. Our scheme has very fewer extra costs of encryption and decryption compared with the

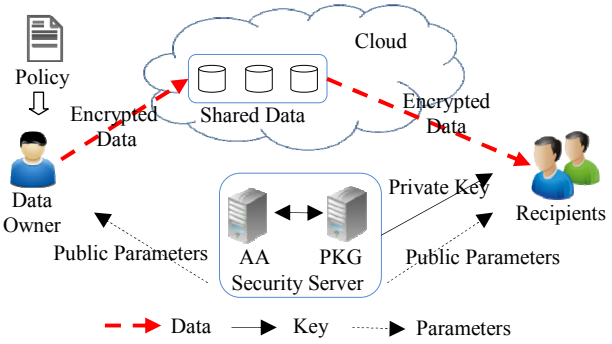

Figure 1. The implementation framework of CP-ABE-HP scheme for outsourced data sharing.

CP-ABE schemes with tree-based access structure, and it can achieve the Chosen-plaintext Attack security under the standard model. The CP-ABE-HP scheme could be a useful scheme in realizing self-contained data protection in cloud computing. For future work, we will implement an efficient CP-ABE-HP mechanism and apply it to some specific cloud storage environments.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No.61170088) and Foundation of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2013ZX-05).

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Advances in Cryptology – EUROCRYPT 2005. vol. 3494, R. Cramer, Ed., ed: Springer Berlin Heidelberg, 2005, pp. 457-473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," presented at the Proceedings of the 13th ACM conference on Computer and communications security, Alexandria, Virginia, USA, 2006, pp. 89-98.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in Security and Privacy, 2007. SP '07. IEEE Symposium on, 2007, pp. 321-334.

[4] K. Frikken, M. Atallah, and J. Li, "Attribute-based access control with hidden policies and hidden credentials," Computers, IEEE Transactions on, vol. 55, 2006, pp. 1259-1270.

[5] S. Yu, K. Ren, and W. Lou, "Attribute-based content distribution with hidden policy," in Secure Network Protocols, 2008. NPSec 2008. 4th Workshop on, 2008, pp. 39-44.

[6] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," presented at the Proceedings of the 6th international conference on Applied cryptography and network security, NewYork, NY, USA, 2008, pp. 111-129.

[7] X. Li, D. Gu, Y. Ren, N. Ding, and K. Yuan, "Efficient ciphertext-policy attribute based encryption with hidden policy," in Internet and Distributed Computing Systems, ed: Springer, 2012, pp. 146-159.

[8] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding CP-ABE," in Information Security Practice and Experience, ed: Springer, 2011, pp. 24-39.

[9] N. Doshi and D. Jinwala, "Hidden access structure ciphertext policy attribute based encryption with constant length ciphertext," in Advanced Computing, Networking and Security, ed: Springer, 2012, pp. 515-523.

[10] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes," in Information Security Practice and Experience. vol. 5451, F. Bao, H. Li, and G. Wang, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 1-12.

[11] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," presented at the Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA, 2007, pp. 456-465.

[12] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded Ciphertext Policy Attribute Based Encryption," in Automata, Languages and Programming. vol. 5126, L. Aceto, I. Damgård, L. Goldberg, M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 579-591.

[13] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, pp. 612-613, 1979.

[14] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," in Public Key Cryptography – PKC 2011. vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 53-70.

[15] A. Kapadia, P. P. Tsang, and S. W. Smith, "Attribute-based publishing with hidden credentials and hidden policies," NDSS'07, pp. 179-192, 2007.

[16] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," Computer Networks, vol. 54, pp. 377-386, 2008.

[17] A. Balu and K. Kuppusamy, "Ciphertext policy attribute based encryption with anonymous access policy," arXiv preprint arXiv:1011.0527, 2010.

[18] S. Yu, "Data sharing on untrusted storage with attribute-based encryption," WORCESTER POLYTECHNIC INSTITUTE, 2010.

[19] A. Balu and K. Kuppusamy, "Privacy Preserving Ciphertext Policy Attribute Based Encryption," in Recent Trends in Network Security and Applications, ed: Springer, 2010, pp. 402-409.

[20] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, 2012, pp. 18-19.

[21] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in Information Security, ed: Springer, 2009, pp. 347-362.

[22] B. Waters, "Efficient identity-based encryption without random oracles," in Advances in Cryptology–EUROCRYPT 2005, ed: Springer, 2005, pp. 114-127.

[23] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in Theory of cryptography, ed: Springer, 2005, pp. 325-341.

[24] A. Beimel, "Secure schemes for secret sharing and key distribution," PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[25] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length," in Information Security Practice and Experience. vol. 5451, F. Bao, H. Li, and G. Wang, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 13-23.