An Efficient and Privacy Preserving CP-ABE Scheme for Internet-Based Collaboration

Jinmiao Wang(✉) and Bo Lang

State Key Laboratory of Software Development Environment,
Beihang University, Beijing 100191, China
{wangjinmiao,langbo}@buaa.edu.cn

Abstract. The development of Internet applications facilitates enterprises and individuals to share information and work together across physical barriers. In such environment, flexible and efficient data-protection methods are required because data are out of the control domain of its owners. By encrypting with an access policy, ciphertext-policy attribute-based encryption (CP-ABE) can simultaneously achieve data encryption and access control, making it an ideal mechanism for data protection in Internet-based environments. However, the existing CP-ABE schemes usually have limitations regarding efficiency and privacy leakage from the access policy. In this paper, we propose a CP-ABE scheme with hidden access policy and fast decryption that improves the decryption efficiency and preserves the privacy of the access policy. In addition, by adopting dual-system encryption methodology, our scheme achieves full security, which is a higher security level in CP-ABE. The performance analysis revealed that the comprehensive capability of our scheme outperforms the existing CP-ABE schemes.

Keywords: Data protection · Attribute-based encryption
Hidden policy · Fast decryption · Fully secure

1 Introduction

With the development of Internet applications and smart devices, people can interact, share information and work together across physical barriers typically exploiting Internet-based environments. Thus, the cooperation among enterprises, public institutions and user communities is evolving into self-organizing and open pattern. For instance, using online social networks to form a collaborative group is very popular at present. During the collaboration, users and institutions tend to outsource their data to an external server on Internet to accomplish data sharing and enable cooperation. In this case, the data objects are out of the control domain of data owners and stored in the untrusted servers, which causes the prominent data security problem. Furthermore, data owners

I. Romdhani et al. (Eds.): CollaborateCom 2017, LNICST 252, pp. 493–507, 2018.
https://doi.org/10.1007/978-3-030-00916-8_46

usually want to share data to some specific users. For example, in e-Health environment patients usually want to share their physical data to doctors in a specific hospital, which can be ensured by access control. Hence, building an effective data protection and access control mechanism for Internet-based collaboration has become a major challenge.

Currently, encryption is the primary mechanism used to ensure data confidentiality. However, traditional public-key encryption cannot achieve efficient and fine-grained access control. Fortunately, the introduction of ciphertext-policy attribute-based encryption (CP-ABE) [1] has made important steps toward solving these problems. In CP-ABE, data are encrypted under an access policy that is specified by the data owner, and a user's private key is generated based on a set of user attributes. If and only if a user's attributes satisfy the access policy can the user decrypt the corresponding ciphertext. Hence, CP-ABE integrates encryption and access control. The access policy in CP-ABE can be expressed based on AND-gate, Tree or Linear Secret-Sharing Scheme (LSSS) matrix. The tree and LSSS structures can express any monotone access policy; hence, both of them are more expressive than the AND-gate. Using its hierarchy, the tree structure allows the data owner to specify a complex access policy in intuitionistic form, thereby delivering a better user experience than LSSS.

In CP-ABE mechanism, not just the data are sensitive but also the access policy, because the access policy may directly contain private information about the protected data and users. For example, through the access policy of a patient's physical data, one can obtain some personal information about the patient, such as the age, the diseases, etc. However, in the original CP-ABE schemes, the access policy is published together with the ciphertext, and anyone who receives the ciphertext can obtain the policy. CP-ABE with hidden access policy presents a good solution to this problem [2–11]. In these schemes, the attributes in the access policy are hidden such that even the legitimate recipient cannot obtain any information about the access policy more than the fact that he can decrypt the ciphertext, which ensures anyone cannot obtain attribute information about the data owner and recipients. However, some existing policy hiding schemes [2–8,10] are based on AND-gate which has limited policy expression ability, and some are proven *selectively* secure [4–9], a weaker security model in which part of the challenged ciphertext description must be declared before the attacker receives the public parameters [12].

In addition, CP-ABE is usually characterized by problems with efficiency. In most existing CP-ABE schemes, the number of pairing operations in decryption, which consumes substantially more CPU time and memory than other operations, increases linearly with the number of attributes involved, which makes the decryption of CP-ABE schemes expensive. To improve the decryption efficiency, many researchers focused on fast decryption [2–5,13,14], i.e., reducing the number of pairing operations in decryption to a constant. However, some of them are based on expressive-ability limited AND-gate, some reveals the access policy, and some are just proven selectively secure. In other words, none of the existing

CP-ABE schemes can achieve fast decryption and policy hiding while remaining policy-expressive and *fully* secure[1].

To address the above limitations, we propose a tree-based CP-ABE with hidden access policy and fast decryption (CP-ABE-HF) in this paper. In the decryption of CP-ABE-HF, the number of pairing operations is reduced to three, which greatly improves the decryption efficiency. To the best of our knowledge, this is the first tree-based CP-ABE scheme that can achieve fast decryption. To preserve the privacy of the access policy, we compute a ciphertext element and record a path in the access tree for each attribute in the system. Thus, no one can obtain what attributes are used in the *real* access policy, even if they are authorized to access the data object. Regarding security, by adopting the dual-system encryption methodology [12], CP-ABE-HF is proven fully secure, thereby overcoming the weakness of the selectively secure model and reaching a higher security level.

The paper is organized as follows. We review the related work in Sect. 2. The preliminaries and background knowledge of our scheme are introduced in Sect. 3. In Sect. 4, we describe the method of policy hiding and propose the CP-ABE scheme with hidden access policy and fast decryption. In Sect. 5, we analysis the security and efficiency of our scheme and compare it with some existing CP-ABE schemes. The paper is concluded in Sect. 6.

2 Related Work

To preserve the access-policy privacy, several CP-ABE schemes with hidden access policy have been proposed. Nishide et al. [6] first constructed an anonymous CP-ABE based on AND-gate, which has limited policy-expression ability. Subsequently, several policy-hiding schemes were proposed with the same access structure [2–5,7,8,10]. Hur [9] proposed a CP-ABE with hidden policies based on the tree structure. However, the access policy can be determined by comparing the index of a user's attributes with the attributes in the ciphertext. Regarding security, the aforementioned schemes are all proven selectively secure. To improve the security, some additional schemes constructed over composite-order groups are proven fully secure. Lai et al. [10] proposed a fully secure and policy-hiding CP-ABE scheme, which is also based on AND-gate. Lai et al. [11] proposed a partially policy-hidden scheme (i.e., the attribute name is revealed while the attribute value is hidden) based on LSSS structure. However, this scheme is inefficient because it adds some redundant components to the ciphertext.

To improve the decryption efficiency of CP-ABE, some researchers focused on reducing the number of pairing operations to a constant. Emura et al. [13] proposed a CP-ABE in which both the ciphertext length and the number of pairing operations are constant. Miyaji et al. [14] proposed a dual-policy ABE that requires four pairing operations to decrypt. Both schemes are based on AND-gate and proven selectively secure.

[1] Full security overcomes the weakness of selective security; i.e., it does not require the attacker to declare the challenged access policy in advance.

There are also some schemes achieved both of policy hidden and fast decryption. The selectively secure schemes proposed in [4,5] reduced the pairing operations to two and four times, respectively, while preserving the policy privacy. Rao and Dutta [2] proposed a fully secure CP-ABE with hidden policy and constant decryption costs. The fully secure scheme proposed by Li et al. [3] also preserved the policy privacy and reduced the number of pairing operations to a constant. However, the aforementioned schemes can only express restricted-access policies; thus, their expression abilities need to be improved.

3 Background Knowledge

3.1 Preliminaries

Composite-Order Bilinear Group. Let $\mathcal{G}$ denote an algorithm that takes as input a security parameter γ and outputs a tuple $(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, e)$, where p_1, p_2, p_3, p_4 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order N, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map such that:

- (Bilinear) $\forall g, h \in \mathbb{G}$ and $x, y \in \mathbb{Z}_N$, it satisfies $e(g^x, h^y) = e(g, h)^{xy}$.
- (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

We require that the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ and the bilinear map e are all computable in polynomial time. Let $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ and $\mathbb{G}_{p_4}$ denote the subgroups of $\mathbb{G}$ with orders p_1, p_2, p_3 and p_4, respectively. Note that if $g_i \in \mathbb{G}_{p_i}$ and $g_j \in \mathbb{G}_{p_j}$ for $i \neq j$, then $e(g_i, g_j) = 1$. If the generator of $\mathbb{G}_{p_i}$ is $g_i (i \in \{1, 2, 3, 4\})$, then every element $h \in \mathbb{G}$ can be expressed as $g_1^{a_1} g_2^{a_2} g_3^{a_3} g_4^{a_4}$ for some values $a_1, a_2, a_3, a_4 \in \mathbb{Z}_N$.

Complexity Assumption. We now present the complexity assumptions that will be used in our scheme. These assumptions are the same as those in [12], and we use them in the group whose order is a product of four primes.

Assumption 1. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{gathered}
(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\gamma), \\
g_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}, \\
D = (\mathbb{G}, g_1, X_3, X_4), \\
T_1 \xleftarrow{R} \mathbb{G}_{p_1p_2}, T_2 \xleftarrow{R} \mathbb{G}_{p_1}.
\end{gathered} \tag{1}$$

The advantage of algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv^1_{\mathcal{G},\mathcal{A}}(\gamma) = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|. \tag{2}$$

Definition 1. *If for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$, $Adv^1_{\mathcal{G},\mathcal{A}}(\gamma)$ is negligible, then we say $\mathcal{G}$ satisfies Assumption 1.*

Assumption 2. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{gathered}
(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\gamma), \\
g_1, X_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, \\
X_3, Y_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}, \\
D = (\mathbb{G}, g_1, X_1X_2, Y_2Y_3, X_3, X_4), \\
T_1 \xleftarrow{R} \mathbb{G}_{p_1p_2p_3}, T_2 \xleftarrow{R} \mathbb{G}_{p_1p_3}.
\end{gathered} \tag{3}$$

The advantage of algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv^2_{\mathcal{G},\mathcal{A}}(\gamma) = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|. \tag{4}$$

Definition 2. *If for any PPT algorithm $\mathcal{A}$, $Adv^2_{\mathcal{G},\mathcal{A}}(\gamma)$ is negligible, then we say $\mathcal{G}$ satisfies Assumption 2.*

Assumption 3. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{gathered}
(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\gamma), \\
g_1 \xleftarrow{R} \mathbb{G}_{p_1}, g_2, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, \\
X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}, \alpha, s \xleftarrow{R} \mathbb{Z}_N, \\
D = (\mathbb{G}, g_1, g_2, g_1^{\alpha}X_2, g_1^{s}Y_2, X_3, X_4), \\
T_1 = e(g_1, g_1)^{\alpha s}, T_2 \xleftarrow{R} \mathbb{G}_T.
\end{gathered} \tag{5}$$

The advantage of algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv^3_{\mathcal{G},\mathcal{A}}(\gamma) = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|. \tag{6}$$

Definition 3. *If for any PPT algorithm $\mathcal{A}$, $Adv^3_{\mathcal{G},\mathcal{A}}(\gamma)$ is negligible, then we say $\mathcal{G}$ satisfies Assumption 3.*

3.2 Background of CP-ABE

Access Tree. Let $\mathcal{T}$ be a tree representing an access structure. Each non-leaf node of the tree represents a threshold operator, which is described by its children and a threshold value. If num_x is the number of children of node x, and k_x is its threshold value, then $1 \leq k_x \leq num_x$. When $k_x = 1$, the threshold is an OR operator, and when $k_x = num_x$, it is an AND operator. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$ [1].

Let $\mathcal{T}$ be an access tree with root r. The subtree of $\mathcal{T}$ rooted at node x is denoted by $\mathcal{T}_x$. Thus, $\mathcal{T}$ is the same as $\mathcal{T}_r$. If a set of attributes ω satisfies the access tree $\mathcal{T}_x$, we denote it as $\mathcal{T}_x(\omega) = 1$. We compute $\mathcal{T}_x(\omega)$ recursively as follows. If x is a non-leaf node, we evaluate $\mathcal{T}_{x'}(\omega)$ for each child x' of node x. $\mathcal{T}_x(\omega)$ returns 1 if and only if at least k_x children return 1. If x is a leaf node, then $\mathcal{T}_x(\omega)$ returns 1 if and only if $att(x) \in \omega$, where $att(x)$ denotes the attribute associated with node x [1].

CP-ABE Algorithms. The CP-ABE consists of the following algorithms [1]:

- **Setup** $(\boldsymbol{U})$. This algorithm takes as input an attribute universe U. It will initialize the system and generate the master key mk and the public key pk.
- **KeyGen** $(\boldsymbol{pk}, \boldsymbol{mk}, \boldsymbol{\omega})$. This algorithm takes as input the public key pk, the master key mk and a user's attribute set ω. It will output a private key sk_ω.
- **Encryption** $(\boldsymbol{pk}, \boldsymbol{M}, \boldsymbol{\mathcal{T}})$. This algorithm takes as input the public key pk, a message M and an access-policy tree $\mathcal{T}$. It will produce a ciphertext $C_\mathcal{T}$.
- **Decryption** $(\boldsymbol{sk_\omega}, \boldsymbol{C_\mathcal{T}})$. The decryption algorithm takes as input a private key sk_ω and a ciphertext $C_\mathcal{T}$. It will output the plaintext M if ω satisfies $\mathcal{T}$.

Security Model. In our scheme, the security under chosen-plaintext attack (CPA) is modeled as a game between a challenger and an adversary. It includes five phases, which are detailed as follows:

- **Setup**. The challenger initializes the system to generate pk and mk. Then, he sends pk to the adversary.
- **Phase 1**. The adversary is allowed to make private key requests for any attribute set ω. The challenger returns sk_ω to the adversary.
- **Challenge**. The adversary sends two equal-length message M_0 and M_1 and two access trees $\mathcal{T}_0^*$ and $\mathcal{T}_1^*$ to the challenger, with the restriction that $\mathcal{T}_0^*$ and $\mathcal{T}_1^*$ cannot be satisfied by any requested attribute set in Phase 1 or contain repeated attributes. The challenger chooses random $\theta \in \{0, 1\}$ and encrypts M_θ with $\mathcal{T}_\theta^*$. Then, the ciphertext $C_{\mathcal{T}_\theta^*}$ is returned to the adversary.
- **Phase 2**. **Phase 1** is repeated with the restriction that none of the requested attribute sets can satisfy $\mathcal{T}_0^*$ or $\mathcal{T}_1^*$ or contain repeated attributes.
- **Guess**. The adversary outputs a guess $\theta' \in \{0, 1\}$.

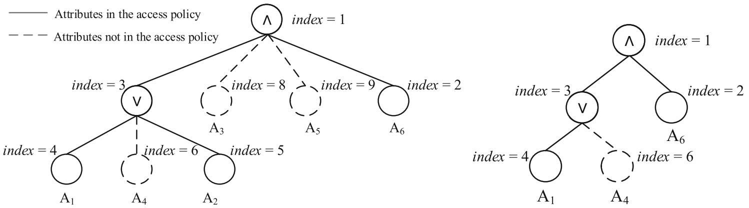
Definition 4. *A CP-ABE scheme with hidden policy is said to be fully secure against CPA if any polynomial-time adversaries have at most a negligible advantage in this security game. The advantage of an adversary is defined as* $\varepsilon = |Pr[\theta' = \theta] - 1/2|$.

Selective security is defined by adding an initialization phase in which the adversary must declare $\mathcal{T}_0^*$ and $\mathcal{T}_1^*$ before receiving pk. In our scheme, we do not impose this restriction on the adversary.

4 Our Constructions

4.1 Hiding the Access Policy

In our construction, the access policy is expressed by an access tree $\mathcal{T}$. To achieve the goal of policy hiding, we randomly choose $index(x) \in \mathbb{Z}_N^*$ for each node x, where $index(x)$ denotes the index of node x in $\mathcal{T}$. Next, for each leaf node x, let $att(x) = A_i$ and record a node path $path_i$ which consists of the index $index(x')$ of each node x' on the path from the root node r to x. For example, the path record of the leaf node associated with attribute A_1 in Fig. 1(a) is $path_1 = \{1, 3, 4\}$.



(a) Record a path for each attribute (b) Reconstruct a tree

Fig. 1. The hiding of access policy

For each attribute that is not in $\mathcal{T}$, we also need to record a node path. The $path_i$ for attribute A_i can be obtained by randomly choosing a non-leaf node z in the tree and implicitly setting A_i to be a dummy child node of z, such as the attributes A_3, A_4 and A_5 which are illustrated as dotted lines in Fig. 1(a). Thus, each of the attributes in the attribute universe has a path record associated with the access tree $\mathcal{T}$. Finally, the path records are sent along with the ciphertext while the access tree $\mathcal{T}$ is discarded, thereby the access policy is hidden. Note that each attribute can only be used once in the access policy because the reuse of an attribute will introduce another path. Thus one can determine that the attributes with more than one path must have appeared in the access policy, which means such attribute is leaked.

In our scheme, the index i for each attribute in the attribute universe is fixed. Hence, with the index i of attributes A_i in the user's attribute set, the decryption algorithm can extract the corresponding $path_i$ from the ciphertext. Then a tree $\mathcal{T}'$ is reconstructed with the records in these paths, and the decryption operation is performed over $\mathcal{T}'$. For example, suppose a user's attribute set is $\omega = \{A_1, A_4, A_6\}$, then the paths $path_1, path_4$ and $path_6$ will be extracted to construct a tree $\mathcal{T}'$, as shown in Fig. 1(b). We can notice that A_4 is not in the access policy, yet it still takes part in the decryption because the user does not know the fact. By performing the decryption algorithm in Sect. 4.2, the ciphertext and private key elements associated with A_4 will be cancelled. Hence, if the user's attributes satisfy the access policy, the user can decrypt successfully without knowing the access policy.

4.2 The CP-ABE-HF Scheme

We construct the fully secure CP-ABE-HF scheme in composite-order groups with order $N = p_1p_2p_3p_4$. Let $\mathbb{G}_{p_i}$ denote the subgroup whose order is prime p_i. The normal operations of the scheme essentially occur in the subgroup $\mathbb{G}_{p_1}$. Private keys are additionally randomized in $\mathbb{G}_{p_3}$, and ciphertexts are additionally randomized in $\mathbb{G}_{p_4}$. The subgroup $\mathbb{G}_{p_2}$ is not used in the real scheme but serves as the semi-functional space in the security proof. We also define the Lagrange coefficient $\ell_{i,V}$ for $i \in \mathbb{Z}_N^*$ and a set, V, of elements in $\mathbb{Z}_N^*$: $\ell_{i,V}(x) = \prod_{j \in V, j \neq i} \frac{x-j}{i-j}$. The CP-ABE-HF scheme includes the following algorithms:

Setup (U). The setup algorithm chooses a bilinear group $\mathbb{G}$ with order $N = p_1p_2p_3p_4$. For each attribute $A_i \in U$ ($1 \leq i \leq n$, where n denotes the number of attributes in the attribute universe U, and the index i for each attribute is fixed), select $h_i \in \mathbb{Z}_N^*$. Then, choose random elements $\alpha, \kappa \in \mathbb{Z}_N^*$ and $g \in \mathbb{G}_{p_1}$. The public key is published as $pk = \{N, g, y = e(g,g)^\alpha, L = g^\kappa, H_i = g^{h_i} (1 \leq i \leq n)\}$, and the master key is $mk = \{\alpha, \kappa\}$.

KeyGen (pk, mk, ω). To generate a private key for a user with attribute set ω, the algorithm chooses random elements $t \in \mathbb{Z}_N^*$ and $R, R_0, \{R_i\}_{A_i \in \omega} \in \mathbb{G}_{p_3}$ and returns the following private key to the user:

$$\begin{aligned} sk_\omega = \{&D = g^{\alpha - \kappa t} R, D_0 = g^t R_0 \\ &\forall A_i \in \omega : D_i = H_i^t R_i\}. \end{aligned} \tag{7}$$

Encryption $(pk, M, \mathcal{T})$. To output the ciphertext of message M encrypted under the access tree $\mathcal{T}$, the algorithm first chooses a polynomial f_x for each node x in $\mathcal{T}$ in the following way, starting with the root node and then proceeding in a top-down manner.

For each node x in the tree, set the degree d_x of the polynomial f_x to be one less than the threshold value k_x; i.e., set $d_x = k_x - 1$. For the root node r, the algorithm chooses a random element s and sets $f_r(0) = s$. Then, it chooses d_r other points of the polynomial f_r randomly to define it. For any other node x, it sets $f_x(0) = f_{parent(x)}(index(x))$ and chooses d_x other points randomly to completely define f_x, where $parent(x)$ indicates the parent of node x. After all of the polynomials are defined, set $\lambda_x = f_x(0)$ for each node x in the tree.

Next, for each attribute A_i in the attribute universe, the node path $path_i$ is recorded using the method detailed in Sect. 4.1. Finally, choose random elements $Z_0, \{Z_i\}_{A_i \in U} \in \mathbb{G}_{p_4}$, and let $att(x) = A_i$, the ciphertext is generated as follows.

$$\begin{aligned} C_\mathcal{T} = \{&E = My^s, E_0 = g^s Z_0, \\ &\forall A_i \in \mathcal{T} : E_i = L^{\lambda_x} H_i^s Z_i, path_i, \\ &\forall A_i \notin \mathcal{T} : E_i = H_i^s Z_i, path_i\}. \end{aligned} \tag{8}$$

We note that for $A_i \in \mathcal{T}$ and $A_i \notin \mathcal{T}$, E_i is properly distributed as a random element of $\mathbb{G}_{p_1p_4}$. Thus, the recipient cannot distinguish E_i of $A_i \in \mathcal{T}$ from E_i of $A_i \notin \mathcal{T}$, indicating that the access policy is fully hidden.

Decryption $(sk_\omega, C_\mathcal{T})$. To decrypt the ciphertext $C_\mathcal{T}$ with sk_ω, the ciphertext component E_i that corresponds to $A_i \in \omega$ should be extracted according to the index i in the attribute universe. Then, an access tree $\mathcal{T}'$ can be reconstructed with the path record $path_i$ in E_i. Next, the decryption algorithm will be executed with tree $\mathcal{T}'$.

First, we define two recursive functions: $DecryptNode\_CT(C_\mathcal{T}, x)$, which takes as inputs the ciphertext $C_\mathcal{T}$ and node x from tree $\mathcal{T}'$, and $DecryptNode\_SK(sk_\omega, x)$, which takes as inputs the private key sk_ω and node x from tree $\mathcal{T}'$, as follows.

If node x is a leaf node, let $att(x) = A_i$ and set:

$$\begin{aligned} &DecryptNode\_CT(C_{\mathcal{T}}, x) = E_i = \begin{cases} L^{\lambda_x} H_i^s Z_i & \text{if } A_i \in \mathcal{T} \\ H_i^s Z_i & \text{if } A_i \notin \mathcal{T} \end{cases} \\ &DecryptNode\_SK(sk_\omega, x) = D_i = H_i^t R_i. \end{aligned} \tag{9}$$

Next, we consider the recursive case when x is an internal node. The functions $DecryptNode\_CT(C_{\mathcal{T}}, x)$ and $DecryptNode\_SK(sk_\omega, x)$ will proceed as follows: For any node y that is the child of x, $DecryptNode\_CT(C_{\mathcal{T}}, y)$ and $DecryptNode\_SK(sk_\omega, y)$ are invoked, and the outputs are stored as F_y and K_y, respectively. Let Q_x be a set of child nodes y that is in $\mathcal{T}$ and Q'_x be a set of y that is not in $\mathcal{T}$. Note that the set $Q_x \cup Q'_x$ denotes all child nodes of x in the reconstructed tree $\mathcal{T}'$. If node y is a leaf node, we compute

$$\begin{aligned} F_x &= \prod_{y \in (Q_x \cup Q'_x)} F_y^{\ell_{y,V_x}(0)} \\ &= \prod_{y \in Q_x} (L^{\lambda_y} H_i^s Z_i)^{\ell_{y,V_x}(0)} \cdot \prod_{y \in Q'_x} (H_i^s Z_i)^{\ell_{y,V_x}(0)} \\ &= \prod_{y \in Q_x} g^{\kappa \lambda_y \cdot \ell_{y,V_x}(0)} \cdot \prod_{y \in (Q_x \cup Q'_x)} H_i^{s \cdot \ell_{y,V_x}(0)} \cdot \prod_{y \in (Q_x \cup Q'_x)} Z_i^{\ell_{y,V_x}(0)} \\ &= g^{\kappa \lambda_x} \cdot \widetilde{F}_{x,1} \cdot \widetilde{F}_{x,2} \end{aligned} \tag{10}$$

and

$$\begin{aligned} K_x &= \prod_{y \in (Q_x \cup Q'_x)} K_y^{\ell_{y,V_x}(0)} \\ &= \prod_{y \in (Q_x \cup Q'_x)} H_i^{t \cdot \ell_{y,V_x}(0)} \cdot \prod_{y \in (Q_x \cup Q'_x)} R_i^{\ell_{y,V_x}(0)} \\ &= \widetilde{K}_{x,1} \cdot \widetilde{K}_{x,2}. \end{aligned} \tag{11}$$

If node y is a non-leaf node, we compute:

$$\begin{aligned} F_x &= \prod_{y \in (Q_x \cup Q'_x)} F_y^{\ell_{y,V_x}(0)} \\ &= \prod_{y \in (Q_x \cup Q'_x)} (g^{\kappa \lambda_y} \cdot \widetilde{F}_{y,1} \cdot \widetilde{F}_{y,2})^{\ell_{y,V_x}(0)} \\ &= \prod_{y \in Q_x} (g^{\kappa \lambda_y})^{\ell_{y,V_x}(0)} \cdot \prod_{y \in (Q_x \cup Q'_x)} \widetilde{F}_{y,1}^{\ell_{y,V_x}(0)} \cdot \prod_{y \in (Q_x \cup Q'_x)} \widetilde{F}_{y,2}^{\ell_{y,V_x}(0)} \\ &= g^{\kappa \lambda_x} \cdot \widetilde{F}_{x,1} \cdot \widetilde{F}_{x,2} \end{aligned} \tag{12}$$

and

$$\begin{aligned} K_x &= \prod_{y \in (Q_x \cup Q'_x)} K_y^{\ell_{y,V_x}(0)} \\ &= \prod_{y \in (Q_x \cup Q'_x)} \widetilde{K}_{y,1}^{\ell_{y,V_x}(0)} \cdot \prod_{y \in (Q_x \cup Q'_x)} \widetilde{K}_{y,2}^{\ell_{y,V_x}(0)} \\ &= \widetilde{K}_{x,1} \cdot \widetilde{K}_{x,2}. \end{aligned} \tag{13}$$

In the above equations, we have $\widetilde{F}^t_{x,1} = \widetilde{K}^s_{x,1}$. The parameter $V_x = \{index(y)|y \in (Q_x \cup Q'_x)\}$, and $\ell_{y,V_x}(0)$ is a Lagrange coefficient which can be computed by everyone who knows the index of attributes in $\mathcal{T}'$.

Now that we have defined the functions *DecryptNode_CT* and *DecryptNode_SK*, the decryption algorithm should first call the functions on the root r of $\mathcal{T}'$. Thus, we obtain

$$A = DecryptNode\_CT(C_{\mathcal{T}}, r) = g^{\kappa s} \cdot \widetilde{F}_{r,1} \cdot \widetilde{F}_{r,2} \tag{14}$$

and

$$B = DecryptNode\_SK(sk_\omega, r) = \widetilde{K}_{r,1} \cdot \widetilde{K}_{r,2}. \tag{15}$$

Then, we compute

$$\begin{aligned} C &= e(A, D_0)/e(E_0, B) \\ &= e\big(g^{\kappa s} \cdot \widetilde{F}_{r,1} \cdot \widetilde{F}_{r,2}, g^t R_0\big)/e\big(g^s Z_0, \widetilde{K}_{r,1} \cdot \widetilde{K}_{r,2}\big) \\ &= e\big(g^{\kappa s}, g^t\big) \cdot e\big(\widetilde{F}_{r,1}, g^t\big) \cdot e\big(\widetilde{F}_{r,2}, g^t\big) \cdot e\big(g^{\kappa s} \cdot \widetilde{F}_{r,1} \cdot \widetilde{F}_{r,2}, R_0\big) \\ &\quad /\big(e\big(g^s, \widetilde{K}_{r,1}\big) \cdot e\big(g^s, \widetilde{K}_{r,2}\big) \cdot e\big(Z_0, \widetilde{K}_{r,1} \cdot \widetilde{K}_{r,2}\big)\big) \\ &= e(g, g)^{\kappa ts}. \end{aligned} \tag{16}$$

Finally, the algorithm returns the plaintext M', where

$$M' = \frac{E}{e(E_0, D) \cdot C} = \frac{Me(g, g)^{\alpha s}}{e(g^s Z_0, g^{\alpha - \kappa t} R) \cdot e(g, g)^{\kappa ts}} = M. \tag{17}$$

If the attributes in ω satisfy the hidden access policy, the recipient can decrypt the ciphertext successfully. To inform the recipient whether the decryption is successful, we use a hybrid encryption method in practice [15]. First, the encryptor picks a random $\varphi \in \mathbb{G}_T$ and derives two uniform and independent b-bit symmetric keys (φ_0, φ_1) from φ. Next, it encrypts the message M using a symmetric encryption scheme under key φ_0 to obtain ciphertext C_0. Our encryption algorithm **Encryption** is used to encrypt φ under access tree $\mathcal{T}$ to obtain ciphertext $C_{\mathcal{T}}$. The final ciphertext consists of $(\varphi_1, C_0, C_{\mathcal{T}})$. In the decryption phase, the recipient first recovers φ' from $C_{\mathcal{T}}$ using our **Decryption** algorithm. Then, it derives (φ'_0, φ'_1) from φ'. If $\varphi'_1 = \varphi_1$, it decrypts C_0 under φ'_0 using the symmetric encryption scheme and outputs the plaintext of message M. Otherwise, it outputs $\perp$. According to [15], the false error probability is approximately $1/2^b$. Thus, the recipient can use φ'_1 to check whether the decryption was successful. Furthermore, the hybrid encryption also greatly improves the efficiency because the message is encrypted using symmetric encryption, which is the most efficient encryption mechanism currently available.

5 Security and Performance Analysis

5.1 Security Analysis

Theorem 1. *If Assumptions 1, 2 and 3 hold, then the proposed CP-ABE-HF scheme is fully secure.*

To prove the security of our scheme, we apply the dual-system encryption methodology in [12]. We first define two structures—the semi-functional ciphertext and semi-functional key—which are not used in the real system, but will be used in our proof. A normal private key can decrypt normal or semi-functional ciphertexts, and a normal ciphertext can be decrypted by normal or semi-functional private keys. However, when decrypting a semi-functional ciphertext with a semi-functional private key, an additional pairing under elements in $\mathbb{G}_{p_2}$ will cause decryption to fail.

Semi-functional Ciphertext. Let g_2 denote a generator of $\mathbb{G}_{p_2}$. To produce a semi-functional ciphertext associated with an access tree $\mathcal{T}$, we first produce a normal ciphertext $C_{\mathcal{T}} = (E, E_0, \{E_i, path_i\})$ with the encryption algorithm. Then, we choose random elements $\kappa', s' \in \mathbb{Z}_N^*$ and a random exponent h_i' for each attribute in the attribute universe. Next, we share the secret s' in the manner detailed in the encryption algorithm and obtain a piece of share λ_x' for each attribute in the access tree. The ciphertext is created as follows:

$$\begin{aligned} &E' = E, E_0' = E_0 \cdot g_2^{s'}, \\ &\forall A_i \in \mathcal{T} : E_i' = E_i \cdot g_2^{\kappa' \lambda_x'} g_2^{s' h_i'}, path_i, \\ &\forall A_i \notin \mathcal{T} : E_i' = E_i \cdot g_2^{s' h_i'}, path_i. \end{aligned} \tag{18}$$

Semi-functional Key. A semi-functional key will take one of two forms. To produce the semi-functional key, we first produce a normal key $sk = (D, D_0, \{D_i\}_{\forall A_i \in \omega})$. Then, we choose random elements $\delta, t' \in \mathbb{Z}_N^*$. The semi-functional key of type 1 is set as follows:

$$D' = D \cdot g_2^{\delta}, D_0' = D_0, \{D_i' = D_i\}_{\forall A_i \in \omega}. \tag{19}$$

The semi-functional key of type 2 is set as follows:

$$D' = D \cdot g_2^{\delta}, D_0' = D_0 \cdot g_2^{t'}, \{D_i' = D_i \cdot g_2^{t' h_i'}\}_{\forall A_i \in \omega}. \tag{20}$$

We will prove the security of our scheme using a hybrid argument over a sequence of games. The first game, Game$_{Real}$, is the real security game (i.e., the ciphertext and all keys are normal). In the next game, Game$_0$, all of the keys will be normal, but the ciphertext will be semi-functional. We let q denote the number of key queries made by the adversary. For k from 1 to q, we define:

Game$_{k,1}$. In this game, the challenge ciphertext is semi-functional, the first $k-1$ keys are semi-functional of type 1, the k^{th} key is semi-functional of type 2, and the remaining keys are normal.

Game$_{k,2}$. In this game, the challenge ciphertext is semi-functional, the first k keys are semi-functional of type 1, and the remaining keys are normal.

For notational purposes, we think of Game$_{0,2}$ as another way of denoting Game$_0$. Note that in Game$_{q,2}$, all keys are semi-functional of type 1. The final game, Game$_{Final}$, is defined to be similar to Game$_{q,2}$, except that the ciphertext is a semi-functional encryption of a random message. Hence, in Game$_{Final}$ the adversary's advantage is 0. To prove Theorem 1, we have the following lemmas:

Lemma 1. *Under Assumption 1, any PPT adversary has at most a negligible advantage in distinguishing between Game*$_{Real}$ *and Game*$_0$*.*

Lemma 2. *Under Assumption 2, any PPT adversary has at most a negligible advantage in distinguishing between Game*$_{k-1,2}$ *and Game*$_{k,1}$*.*

Lemma 3. *Under Assumption 2, any PPT adversary has at most a negligible advantage in distinguishing between Game*$_{k,1}$ *and Game*$_{k,2}$*.*

Lemma 4. *Under Assumption 3, any PPT adversary has at most a negligible advantage in distinguishing between Game*$_{q,2}$ *and Game*$_{Final}$*.*

Due to the space limitation, we omit the formal proofs of Lemmas 1–4, which will be given in the full version of the paper. Through Lemmas 1–4, we can prove that Game$_{Real}$ is indistinguishable from Game$_{Final}$. Therefore, we can conclude that the adversary's advantage in breaking the CP-ABE-HF scheme (i.e., Game$_{Real}$) is negligible, which completes the proof of Theorem 1.

5.2 Efficiency Analysis

An overview comparison of our scheme with some existing CP-ABE schemes is presented in Table 1. Our scheme achieves policy hiding and fast decryption simultaneously. The access policy is specified based on the tree structure, which is as expressive as LSSS and more expressive than AND-gate. Regarding security, our scheme is proven *fully* secure in the standard model. The comparison

Table 1. An overview comparison of our scheme with other CP-ABE schemes.

Scheme	Access structure	Policy hidden	Fast decryption	Security
EMN+09 [13]	AND-gate	No	Yes	Selective
MT12 [14]	AND-gate	No	Yes	Selective
NYO08 [6]	AND-gate	Yes	No	Selective
ZHW15 [7]	AND-gate	Yes	No	Selective
LWZ+16 [8]	AND-gate	Yes	No	Selective
H13 [9]	Tree	Yes	No	Selective
DJ12 [4]	AND-gate	Yes	Yes	Selective
PJ14 [5]	AND-gate	Yes	Yes	Selective
LDL12 [11]	LSSS	Yes	No	Full
LDL11 [10]	AND-gate	Yes	No	Full
RD13 [2]	AND-gate	Yes	Yes	Full
LGR+12 [3]	AND-gate	Yes	Yes	Full
CP-ABE-HF (ours)	Tree	Yes	Yes	Full

Table 2. Efficiency comparison of CP-ABE schemes with hidden access policy and full security.

Scheme	Structure	Encryption	KeyGen	Decryption
LGR+12 [3]	AND	$2\mathbb{G} + \mathbb{G}_T$	$4\mathbb{G}$	$2C_e$
RD13 [2]	AND	$2\mathbb{G} + \mathbb{G}_T$	$3\mathbb{G}$	$2C_e$
LDL11 [10]	AND	$(\hat{n}+1)\mathbb{G} + \mathbb{G}_T$	$(\hat{n}+1)\mathbb{G}$	$(\hat{n}+1)C_e$
LDL12 [11]	LSSS	$(8\hat{t}+2)\mathbb{G} + 2\mathbb{G}_T$	$(2\hat{n}+3)\mathbb{G}$	$(4\hat{t}+2)C_e + 2\hat{t}\mathbb{G}_T$
CP-ABE-HF (ours)	Tree	$(n+\hat{t}+1)\mathbb{G} + \mathbb{G}_T$	$(\|\omega\|+2)\mathbb{G}$	$3C_e + 2\|\omega\|\mathbb{G}_T$

Note: $\mathbb{G}$ and $\mathbb{G}_T$ represent the exponentiations on groups of $\mathbb{G}$ and $\mathbb{G}_T$, respectively. C_e denotes the pairing operation. n denotes the number of attributes in the attribute universe. $\hat{n}$ denotes the number of attribute categories in the AND-gate based schemes. $\hat{t}$ denotes the number of attribute in an access structure. $|\omega|$ is the number of attributes associated with a user.

indicates that our scheme is superior to the existing CP-ABE schemes because it is the first CP-ABE scheme that has all of the following features: hidden policy, fast decryption, expressivity and full security.

The performance comparisons of our scheme with other policy-hiding and fully secure CP-ABE schemes are shown in Table 2. Although LGR+12 and RD13 are more efficient than our scheme, their access policy is expressed by AND-gate which has restricted expression ability. Based on the comparisons, we can conclude that our scheme is more suitable for the Internet-based collaboration environments, where data protection and fine-grained access control are required, and the access policy contains sensitive information.

6 Conclusions

In this paper, focusing on the data-protection problem in Internet-based collaboration, we propose an efficient and privacy-preserving scheme based on CP-ABE, i.e., CP-ABE-HF. CP-ABE-HF improves the efficiency of decryption by reducing the number of pairing operations to three, regardless of how complex the access policy is. The privacy of the access policy is also preserved so that no one can obtain the access policy after encryption, which ensures the attribute information of data owners and recipients will not be leaked. By adopting the dual-system encryption methodology, CP-ABE-HF is proven fully secure. The performance analysis indicates that CP-ABE-HF outperforms the existing CP-ABE schemes in terms of its comprehensive capability, because it is the first CP-ABE to simultaneously achieve policy hidden, fast decryption, expressivity and full security. In the future, it would be interesting to construct a fully secure CP-ABE with policy hidden and fast decryption over prime-order groups of which efficiency is higher than the composite-order groups.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant No. 61170088) and the Foundation of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2015ZX-05).

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE, Washington, DC (2007)
2. Rao, Y.S., Dutta, R.: Recipient anonymous ciphertext-policy attribute based encryption. In: Bagchi, A., Ray, I. (eds.) ICISS 2013. LNCS, vol. 8303, pp. 329–344. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45204-8_25
3. Li, X., Gu, D., Ren, Y., Ding, N., Yuan, K.: Efficient ciphertext-policy attribute based encryption with hidden policy. In: Xiang, Y., Pathan, M., Tao, X., Wang, H. (eds.) IDCS 2012. LNCS, vol. 7646, pp. 146–159. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34883-9_12
4. Doshi, N., Jinwala, D.: Hidden access structure ciphertext policy attribute based encryption with constant length ciphertext. In: Thilagam, P.S., Pais, A.R., Chandrasekaran, K., Balakrishnan, N. (eds.) ADCONS 2011. LNCS, vol. 7135, pp. 515–523. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29280-4_60
5. Padhya, M., Jinwala, D.: A novel approach for searchable CP-ABE with hidden ciphertext-policy. In: Prakash, A., Shyamasundar, R. (eds.) ICISS 2014. LNCS, vol. 8880, pp. 167–184. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13841-1_10
6. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 111–129. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_7
7. Zhou, Z., Huang, D., Wang, Z.: Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption. IEEE Trans. Comput. **64**(1), 126–138 (2015)
8. Li, J., Wang, H., Zhang, Y., Shen, J.: Ciphertext-policy attribute-based encryption with hidden access policy and testing. KSII Trans. Internet Inf. Syst. **10**(7), 3339–3352 (2016)
9. Hur, J.: Attribute-based secure data sharing with hidden policies in smart grid. IEEE Trans. Parallel Distrib. Syst. **24**(11), 2171–2180 (2013)
10. Lai, J., Deng, R.H., Li, Y.: Fully secure cipertext-policy hiding CP-ABE. In: Bao, F., Weng, J. (eds.) ISPEC 2011. LNCS, vol. 6672, pp. 24–39. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21031-0_3
11. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: 7th ACM Symposium on Information, Computer and Communications Security, pp. 18–19. ACM, New York (2012)
12. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_12
13. Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.: A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 13–23. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00843-6_2

14. Miyaji, A., Tran, P.V.X.: Constant-ciphertext-size dual policy attribute based encryption. In: Xiang, Y., Lopez, J., Kuo, C.-C.J., Zhou, W. (eds.) CSS 2012. LNCS, vol. 7672, pp. 400–413. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35362-8_30
15. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_29