Expressive CP-ABE with Partially Hidden Access Structures

Junzuo Lai
School of Information Systems
Singapore Management University
junzuolai@smu.edu.sg

Robert H. Deng
School of Information Systems
Singapore Management University
robertdeng@smu.edu.sg

Yingjiu Li
School of Information Systems
Singapore Management University
yjli@smu.edu.sg

ABSTRACT

In a traditional ciphertext-policy attribute-based encryption (CP-ABE) scheme, an access structure, also refereed to as ciphertext-policy, is sent along with a ciphertext explicitly, and anyone who obtains a ciphertext can know the access structure associated with the ciphertext. In certain applications, access structures contain sensitive information and must be protected from everyone except the users whose private key attributes satisfy the access structures.

In this paper, we first propose a new model for CP-ABE with partially hidden access structures. In our model, each attribute consists of two parts: an attribute name and its value; if the private key attributes of a user do not satisfy the access structure associated with a ciphertext, the specific attribute values of the access structure are hidden, while other information about the access structure is public.

Based on the CP-ABE scheme proposed by Lewko et al. [14] recently, we then present an efficient construction of CP-ABE with partially hidden access structures. Compared to previous works in this field, our construction is more flexible and expressive and is proven fully secure in the standard model.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public Key Cryptosystems; H.2.7 [**Database Administration**]: Security, Integrity, and Protection

General Terms

Design, Security

Keywords

Ciphertext-Policy Attribute-Based Encryption, Partially Hidden Access Structure, Dual System Encryption

1. INTRODUCTION

Many distributed applications require complex access- control mechanisms, where access decisions depend on attributes of protected data and access control policies assigned to users, or data owners can establish specific access control policies on who can decrypt the protected data based on users' attributes. Sahai and Waters [26] addressed this issue by introducing the concept of attribute-based encryption (ABE). ABE enables public key based one-to-many encryption and is envisioned as a promising cryptographic primitive for realizing scalable and fine-grained access control systems. There are two kinds of ABE schemes [11], key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) schemes. This paper, our concern is on the latter.

In a CP-ABE scheme [2], every ciphertext is associated with an access structure on attributes, and every user's secret key is associated with a set of attributes. A user will be able to decrypt a ciphertext only if the set of attributes associated with the user's private key satisfies the access structure associated with the ciphertext. In traditional CP-ABE schemes [2, 9, 14, 30], an access structure is sent along with a ciphertext explicitly; therefore anyone who obtains the ciphertext is able to know the associated access structure. However, this property is not appropriate for certain applications where access policies contain sensitive information.

Consider the following cloud data storage scenario where a data owner intends to outsource his data to the cloud and wants to establish specific access control policies on who can access the data. Before outsourcing his data, the data owner encrypts it in order to prevent leakage of sensitive information to the cloud service provider since the cloud is usually operated by commercial firms outside of the data owner's trusted domain. Figure 1 depicts the system architecture of a cloud storage for healthcare information. In healthcare, it must meet the requirements of Health Insurance Portability and Accountability Act (HIPAA) for any use or disclosure of protected healthcare information; therefore, there is no option but to keep medical data confidential against cloud storage servers. Suppose that a data owner intends to outsource a medical record to the cloud and specifies that the medical record can only be accessed by a cardiologist in University Park Hospital or by a patient with social security number 123-45-6789. The data owner encrypts the record using a CP-ABE scheme in order to keep it confidential from the cloud service provider. If the data owner uses a traditional CP-ABE scheme to encrypt the medical record, everyone including the cloud service provider is able to know the access policy associated with the ciphertext, and can in-

ASIACCS '12, May 2–4, 2012, Seoul, Korea.

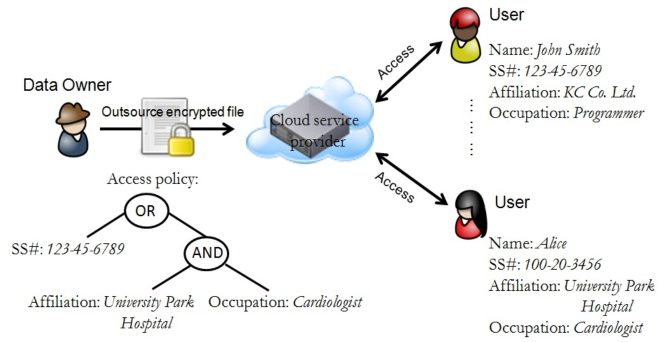

Figure 1: An example of cloud storage system architecture

fer that someone with social security number 123-45-6789 suffers from a heart problem. This is clearly not acceptable and shows the necessity of hiding the access policies from prying eyes in certain applications.

One can construct CP-ABE with hidden access structures from attribute-hiding Inner-product Predicate Encryption (IPE) [12]. Predicate Encryption (PE) was presented by Katz, Sahai and Waters [12] as a generalized (fine-grained) notion of encryption that covers CP-ABE. In a PE scheme, secret keys correspond to predicates and ciphertexts are associated with a set of attributes; a secret key SK_f corresponding to a predicate f can be used to decrypt a ciphertext associated with an attribute set I if and only if $f(I) = 1$. Katz, Sahai, and Waters [12] also introduced the idea of *attribute-hiding*, a security notion for PE that is stronger than the basic security requirement of *payload-hiding*. Roughly speaking, *attribute-hiding* requires that a ciphertext conceal the associated attributes as well as the plaintext, while *payload-hiding* only requires that a ciphertext conceal the plaintext. The special case of inner product predicates is obtained by having each attribute correspond to a vector $\vec{x}$ and each predicate $f_{\vec{v}}$ correspond to a vector $\vec{v}$, where $f_{\vec{v}}(\vec{x}) = 1$ iff $\vec{x} \cdot \vec{v} = 0$. ($\vec{x} \cdot \vec{v}$ denotes the standard inner-product.)

As mentioned in [14], in order to use inner product predicates for CP-ABE, access structures must be written in CNF or DNF form, which can cause a *superpolynomial* blowup in size for arbitrary access structures. Since it is extremely inefficient to implement CP-ABE schemes with fully hidden access structures derived from attribute-hiding IPE, we investigate how to trade off fully hidden access structures for the efficiency of CP-ABE.

1.1 Our Contributions

In many applications, specific attribute values carry much more sensitive information than the generic attribute names. In Figure 1, "Cardiologist" and "123-45-6789" are more sensitive than "Occupation" and "SS#", respectively. This observation motivates us to consider a new model of CP-ABE with partially hidden access structures. In this model, each attribute includes two parts: attribute name and its value; if the set of attributes associated with a user's private key does not satisfy the access structure associated with a ciphertext, attribute values in the access structure are hidden, while other information, such as attribute names, about the access structure is public. In the above-mentioned example, if the data owner uses a CP-ABE scheme under this new model to encrypt his medical record, anyone obtaining the ciphertext only knows the following information about the access policy:

$$\mathrm{SS\#}:\ \ast\ \mathsf{OR}\ (\mathrm{Affiliation}:\ \ast\ \mathsf{AND}\ \mathrm{Occupation}:\ \ast),$$

while the sensitive attribute values, such as"123-45-6789", "University Park Hospital" and "Cardiologist", are hidden from the public. Figure 2 shows graphically this example of partially hidden access structure.

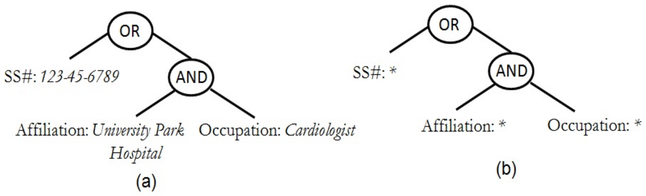

Figure 2: An access structure (a) and the corresponding partially hidden access structure (b)

Based on the CP-ABE scheme proposed by Lewko et al. [14] recently, we present an efficient construction of CP-ABE with partially hidden access structures. In a CP-ABE scheme, if the access structure associated with a ciphertext is fully hidden, a user is not able to know which attribute set satisfies the access structure, and this makes decryption difficult. However, in the proposed CP-ABE with partially hidden access structures, we avoid the problem by adding some redundant components to a ciphertext, where if the private key attributes of a user satisfy the access structure associated with the ciphertext, the user is able to decide which attribute set satisfies the access structure using the redundant components of the ciphertext. Our scheme can handle any access structure that can be expressed as a Linear Secret Sharing Scheme (LSSS), and its ciphertext size scales *linearly* with the complexity of the access structure. We prove that the proposed scheme is fully secure in the standard model using the dual system encryption methodology in [29].

There are a few other works [22, 19, 13] on CP-ABE with partially hidden access structures. However, their schemes only support restricted access structures, which can be expressed as AND gates on multi-valued attributes with wildcards. Compared to these schemes, our scheme is more flexible and expressive. An overview comparing our CP-ABE scheme to those of other CP-ABE schemes with hidden access structures is given in Table 1. The table shows that our scheme is superior to the existing ones in the area of CP-ABE with partially hidden access structures since it can handle the most expressive access structures and is fully secure in the standard model. Therefore, our proposed scheme is most suitable for outsourcing data with sensitive attribute values in access control policies.

1.2 Related Work

In this section, we summarize the major related works in the areas of ABE, KP-ABE, PE, CP-ABE, CP-ABE with partially hidden access structures, and dual system encryption technology.

Attribute-Based Encryption (ABE). The notion of ABE was first introduced by Sahai and Waters as an application

Scheme	Anonymity of access structures	Expressiveness of access structures	Security	Ciphertext size
CP-ABE [14]	no	LSSS	fully secure	linear
IPE* [14]	fully hidden	inner product predicates	fully secure	linear
[22, 19]	partially hidden	AND-gates on multi-valued attributes with wildcards	selectively secure	linear
[13]	partially hidden	AND-gates on multi-valued attributes with wildcards	fully secure	linear
Ours	partially hidden	LSSS	fully secure	linear

Table 1: Comparison of CP-ABE schemes, where "linear" means that the size of ciphertext scales *linearly* with the complexity of the access structure. *In a CP-ABE scheme with fully hidden access structure which is derived from attribute hiding IPE, the access structure must be converted to an inner-product predicate and this causes a *superpolynomial* blowup in ciphertext size.

of their fuzzy identity-based encryption (IBE) scheme [26], where both ciphertexts and secret keys are associated with sets of attributes. The decryption of a ciphertext is enabled if and only if the attribute set for the ciphertext and the attribute set for the secret key overlap by at least a fixed threshold value d.

KP-ABE. Goyal et al. [11] formulated two complimentary forms of ABE: KP-ABE and CP-ABE. In a CP-ABE scheme, decryption keys are associated with sets of attributes and ciphertexts are associated with access structures. In a KP-ABE scheme, the situation is reversed: decryption keys are associated with access structures while ciphertexts are associated with sets of attributes. There exists a general method to transform KP-ABE to CP-ABE [10]. In terms of the expressive power of access structures, Goyal et al. [11] presented the first KP-ABE supporting monotonic access structures. To enable more flexible access control policy, Ostrovsky et al. [25] presented a KP-ABE system that supports the expression of non-monotone formulas in key policies. The problem of building KP-ABE systems with multiple authorities was investigated in [7, 20, 8]. Recently, Lewko and Waters [18] proposed a KP-ABE scheme which is "unbounded" in the sense that the public parameters do not impose additional limitations on the functionality of the scheme.

Predicate Encryption (PE). In this paragraph, we give a brief introduction about the work on PE since CP-ABE can be derived from inner-product PE. The notion of PE was introduced by Katz et al. [12]. They also proposed the first inner-product PE. Shi and Waters [28] presented a delegation mechanism for a class of PE, in which the admissible predicates of the system are more limited than inner-product predicates. Okamota and Takashima [23] presented a (hierarchical) delegation mechanism for an inner-product PE scheme. Shen et al. [27] introduced a new security notion of PE called predicate privacy and proposed a symmetric-key inner-product PE, which achieves both plaintext privacy and predicate privacy. These schemes were only proven selectively secure. Lweko et al. [14] proposed the first fully secure inner-product PE. Okamota and Takashima [24] presented a fully secure PE for a wide class of admissible predicates, which are specified by non-monotone access structures combined with inner-product predicates.

CP-ABE. The first CP-ABE construction proposed by Bethencourt et al. [2] is proven secure under the generic group model. Later, Cheung and Newport [9] proposed an CP-ABE scheme that is secure under the standard model; however, the access structures in this scheme are restricted to AND of different attributes. Recently, secure and expressive CP-ABE schemes [30, 14] were proposed. CP-ABE schemes with multiple authorities were also studied in [21, 17].

CP-ABE with Partially Hidden Access Structures. The notion of CP-ABE with partially hidden access structures was introduced by Nishide et al. [22], where the admissible access structures are expressed as AND gates on multi-valued attributes with wildcards. Li et al. [19] followed their work and studied the problem of user accountability. All these schemes are proven to be *selectively* secure only, which is a weak security model analogous to the selective-ID model [5, 3] in IBE schemes. Recently, Lai et al. [13] proposed a fully secure (cf. selectively secure) CP-ABE scheme with partially hidden access structures; however, their scheme only supports restricted access structures as in [22, 19]. Moving one step forward, we propose a fully secure CP-ABE scheme with partially hidden access structures that can be expressed as an LSSS, which is more flexible and expressive than previous works [22, 19, 13].

Dual System Encryption Methodology. The dual system encryption methodology, introduced by Waters in [29], will be used in the security proofs of our construction. This methodology has been leveraged to obtain constructions of fully secure (H)IBE from simple assumptions [29], fully secure (H)IBE with short ciphertexts [16], fully secure (H)IBE and ABE with leakage resilience [15], fully secure ABE and inner-product PE [14, 24].

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we review some standard notations and cryptographic definitions. In Section 3, we describe the security model for CP-ABE with partially hidden access structures and propose a concrete construction. Details of the security proofs of the proposed construction are given in the Appendix. We state our conclusion in Section 4.

2. PRELIMINARIES

If S is a set, then $s \overset{\$}{\leftarrow} S$ denotes the operation of picking an element s uniformly at random from S. Let $\mathbb{N}$ denote the set of natural numbers. If $\lambda \in \mathbb{N}$ then 1^λ denotes the string of λ ones. Let $z \leftarrow \mathsf{A}(x, y, \ldots)$ denote the operation of running an algorithm A with inputs $(x, y, \ldots)$ and output z. A function $f(\lambda)$ is *negligible* if for every $c > 0$ there exists a λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

2.1 Access Structures

Definition 1 (Access Structure [1]). *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, 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 $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.*

In our context, attributes play the role of parties and we restrict our attention to monotone access structures. It is possible to (inefficiently) realize general access structures using our techniques by treating the negation of an attribute as a separate attribute.

2.2 Linear Secret Sharing Schemes

Our construction will employ linear secret-sharing schemes (LSSS). We use the definition adapted from [1]:

Definition 2 (Linear Secret-Sharing Schemes). *A secret sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if*

1. *The shares for each party form a vector over $\mathbb{Z}_p$.*

2. *There exists a matrix $\mathbf{A}$ with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, \ell$, the i^{th} row of $\mathbf{A}$ is labeled by a party $\rho(i)$ (ρ is a function from $\{1, \ldots, \ell\}$ to $\mathcal{P}$). When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $\mathbf{A}v$ is the vector of ℓ shares of the secret s according to Π. The share $(\mathbf{A}v)_i$ belongs to party $\rho(i)$.*

It is shown in [1] that every linear secret-sharing scheme according to the above definition also enjoys the linear reconstruction property, defined as follows. Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, \ldots, \ell\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Let A_i denotes the i^{th} row of $\mathbf{A}$, we have $\sum_{i \in I} \omega_i A_i = (1, 0, \ldots, 0)$. These constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generation matrix $\mathbf{A}$ [1]. Note that, for unauthorized sets, no such constants $\{\omega_i\}$ exist.

Boolean Formulas Access structures might also be described in terms of monotonic boolean formulas. Using standard techniques [1] one can convert any monotonic boolean formula into an LSSS representation. We can represent the boolean formula as an access tree. An access tree of ℓ nodes will result in an LSSS matrix of ℓ rows. We refer the reader to the appendix of [17] for a discussion on how to perform this conversion.

2.3 Ciphertext-Policy Attribute-Based Encryption

A CP-ABE scheme consists of the following four algorithms:

$\mathsf{Setup}(1^\lambda, U)$ takes as input a security parameter λ and the attribute universe description U. It outputs the public parameters PK and a master secret key MSK.

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathcal{S})$ takes as input the public parameters PK, the master secret key MSK and a set of attributes $\mathcal{S}$. It outputs a secret key $\mathsf{SK}_\mathcal{S}$.

$\mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A})$ takes as input the public parameters PK, a message M and an access structure $\mathbb{A}$. It outputs a ciphertext C.

$\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_\mathcal{S}, C)$ takes as input the public parameters PK, a secret key $\mathsf{SK}_\mathcal{S}$ and a ciphertext C. It outputs a message M.

Let $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, U)$, $\mathsf{SK}_\mathcal{S} \leftarrow \mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathcal{S})$, $C \leftarrow \mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A})$. For correctness, we require the following to hold:

1. If the set $\mathcal{S}$ of attributes satisfies the access structure $\mathbb{A}$, then $M \leftarrow \mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_\mathcal{S}, C)$;

2. Otherwise, with overwhelming probability, $\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_\mathcal{S}, C)$ outputs a random message.

2.4 Composite Order Bilinear Groups

We will construct our scheme in composite order bilinear groups whose order is the product of four distinct primes. Composite order bilinear groups were first introduced in [4].

Let $\mathcal{G}$ be an algorithm that takes as input a security parameter 1^λ and outputs a tuple $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$, where p_1, p_2, p_3, p_4 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1 p_2 p_3 p_4$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that

1. (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$;

2. (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

We further require that multiplication in $\mathbb{G}$ and $\mathbb{G}_T$, as well as the bilinear map e, are computable in time polynomial in λ. We use $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}, \mathbb{G}_{p_4}$ to denote the subgroups of $\mathbb{G}$ having order p_1, p_2, p_3, p_4, respectively. Observe that $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$. Note also that if $g_1 \in \mathbb{G}_{p_1}$ and $g_2 \in \mathbb{G}_{p_2}$ then $e(g_1, g_2) = 1$. A similar rule holds whenever e is applied to elements in distinct subgroups.

We now state the complexity assumptions we use. Assumptions 1, 2 and 3 are the same assumptions used in [14], and we use it in the group whose order is a product of four primes. Assumption 4 was used in [6].

Assumption 1. *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1 p_2 p_3 p_4,$$

$$g \xleftarrow{\$} \mathbb{G}_{p_1},\ X_3 \xleftarrow{\$} \mathbb{G}_{p_3},\ X_4 \xleftarrow{\$} \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_3, X_4),$$

$$T_1 \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2},\ T_2 \overset{\$}{\leftarrow} \mathbb{G}_{p_1}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 is defined as

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

DEFINITION 3. *we say $\mathcal{G}$ satisfies Assumption 1 if for any polynomial time algorithm $\mathcal{A}$, $Adv^1_{\mathcal{A}}$ is negligible.*

ASSUMPTION 2. *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$

$$g, X_1 \overset{\$}{\leftarrow} \mathbb{G}_{p_1},\ X_2, Y_2 \overset{\$}{\leftarrow} \mathbb{G}_{p_2},\ X_3, Y_3 \overset{\$}{\leftarrow} \mathbb{G}_{p_3},\ X_4 \overset{\$}{\leftarrow} \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_1X_2, Y_2Y_3, X_3, X_4),$$

$$T_1 \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3},\ T_2 \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 is defined as

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

DEFINITION 4. *we say $\mathcal{G}$ satisfies Assumption 2 if for any polynomial time algorithm $\mathcal{A}$, $Adv^2_{\mathcal{A}}$ is negligible.*

ASSUMPTION 3. *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$

$$\alpha, s \in \mathbb{Z}_N,\ g \overset{\$}{\leftarrow} \mathbb{G}_{p_1},$$

$$g_2, X_2, Y_2 \overset{\$}{\leftarrow} \mathbb{G}_{p_2},\ X_3 \overset{\$}{\leftarrow} \mathbb{G}_{p_3},\ X_4 \overset{\$}{\leftarrow} \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g^\alpha X_2, g^s Y_2, X_3, X_4),$$

$$T_1 = e(g, g)^{\alpha s},\ T_2 \overset{\$}{\leftarrow} \mathbb{G}_T.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 is defined as

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

DEFINITION 5. *we say $\mathcal{G}$ satisfies Assumption 3 if for any polynomial time algorithm $\mathcal{A}$, $Adv^3_{\mathcal{A}}$ is negligible.*

ASSUMPTION 4. *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$

$$t', r' \in \mathbb{Z}_N,\ g, h \overset{\$}{\leftarrow} \mathbb{G}_{p_1}, g_2, X_2, A_2, B_2, D_2 \overset{\$}{\leftarrow} \mathbb{G}_{p_2},$$

$$X_3 \overset{\$}{\leftarrow} \mathbb{G}_{p_3},\ X_4, Z, A_4, D_4 \overset{\$}{\leftarrow} \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g^{t'}B_2, h^{t'}Y_2, X_3, X_4, hZ, g^{r'}D_2D_4),$$

$$T = h^{r'}A_2A_4,\ T_2 \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 4 is defined as

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

DEFINITION 6. *we say $\mathcal{G}$ satisfies Assumption 4 if for any polynomial time algorithm $\mathcal{A}$, $Adv^4_{\mathcal{A}}$ is negligible.*

3. CP-ABE WITH PARTIALLY HIDDEN ACCESS STRUCTURES

In this section, we first describe the security model for CP-ABE with partially hidden access structures. Then, based on the CP-ABE scheme proposed by Lewko et al. [14], we propose a new CP-ABE scheme, which satisfies the security definition of partially hidden access structures.

Similar to the scheme in [14], our proposed CP-ABE scheme has the restriction that each attribute name can only be used once in an access formula, which is called one-use CP-ABE. We can obtain a secure CP-ABE scheme with partially hidden access structures where attribute names are used multiple times (up to a constant number of uses fixed at setup) from a one-use scheme by applying the generic transformation given in Lewko et al. [14]. While the transformation does incur some cost in key size, it does not increase the size of the ciphertext.

Our construction supports arbitrary monotone access formulas. As in [14], we express access formulas by an LSSS matrix $\mathbf{A}$ over the attributes in the system, but with a significant difference. In our construction, each attribute includes two parts: attribute name and its value. Without loss of generality, we assume that there are n categories of attributes and every user has n attributes with each attribute belonging to a different category. For notational purposes, let i denote the attribute name of the i^{th} category attribute. A user's attribute set $\mathcal{S}$ is parsed as $(s_1, \ldots, s_n)$, where $s_i \in \mathbb{Z}_N$ is the value of attribute i. We express an access formula by $(\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is $\ell \times n$ share-generating matrix, ρ is a map from each row of $\mathbf{A}$ to an attribute name (i.e., ρ is a function from $\{1, \ldots, \ell\}$ to $\{1, \ldots, n\}$), $\mathcal{T}$ can be parsed as $(t_{\rho(1)}, \ldots, t_{\rho(\ell)})$ and $t_{\rho(i)}$ is the value of attribute $\rho(i)$ specified by the access formula.

Using our notations, a user's attribute set $\mathcal{S} = (s_1, \ldots, s_n)$ satisfies an access formula $(\mathbf{A}, \rho, \mathcal{T})$ if and only if there exist $\mathcal{I} \subseteq \{1, \ldots, \ell\}$ and constants $\{\omega_i\}_{i \in \mathcal{I}}$ such that

$$\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0) \text{ and } s_{\rho(i)} = t_{\rho(i)} \text{ for } \forall i \in \mathcal{I},$$

where A_i denotes the i^{th} row of $\mathbf{A}$. We also say that $\mathcal{I} \subseteq \{1, \ldots, \ell\}$ satisfies $(\mathbf{A}, \rho)$ if there exist constants $\{\omega_i\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$. We define $\mathrm{I}_{\mathbf{A},\rho}$ as the set of minimum subsets of $\{1, \ldots, \ell\}$ that satisfies $(\mathbf{A}, \rho)$. By "minimum", we mean the subset cannot become smaller while still satisfying $(\mathbf{A}, \rho)$.

Note that, in our construction to be presented below, the specific attribute values (i.e., $\mathcal{T}$) of an access formula $(\mathbf{A}, \rho, \mathcal{T})$ is hidden, while other information about the access formula (i.e., $(\mathbf{A}, \rho)$ is sent along with the ciphertext explicitly.

3.1 Security Model for CP-ABE with Partially Hidden Access Structures

We now give the security model for CP-ABE with partially hidden access structures, described as a security game between a challenger and an adversary $\mathcal{A}$. The game proceeds as follows:

Setup The challenger runs $\mathsf{Setup}(1^\lambda, U)$ to obtain the public parameters PK and a master secret key MSK. It gives the public parameters PK to the adversary $\mathcal{A}$ and keeps MSK to itself.

Query phase 1 The adversary $\mathcal{A}$ adaptively queries the challenger for secret keys corresponding to sets of attributes $\mathcal{S}_1, \ldots, \mathcal{S}_q$. In response, the challenger runs $\mathsf{SK}_{\mathcal{S}_i} \leftarrow \mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathcal{S}_i)$ and gives the secret key $\mathsf{SK}_{\mathcal{S}_i}$ to $\mathcal{A}$, for $1 \leq i \leq q$.

Challenge The adversary $\mathcal{A}$ submits two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0)$, $(\mathbf{A}, \rho, \mathcal{T}_1)$, subject to the restriction that, $(\mathbf{A}, \rho, \mathcal{T}_0)$ and $(\mathbf{A}, \rho, \mathcal{T}_1)$ cannot be satisfied by any of the queried attribute sets. The challenger selects a random bit $\beta \in \{0, 1\}$, sets $C = \mathsf{Encrypt}(\mathsf{PK}, M_\beta, (\mathbf{A}, \rho, \mathcal{T}_\beta))$ and sends C to the adversary as its challenge ciphertext.

Note that, the LSSS matrix $\mathbf{A}$ and ρ are the same in the two access structures provided by the adversary. In a CP-ABE scheme with partially hidden access structures, one can distinguish the ciphertexts if the associated access structures have different $(\mathbf{A}, \rho)$, since $(\mathbf{A}, \rho)$ is sent along with the ciphertext explicitly.

Query phase 2 The adversary continues to adaptively query the challenger for secret keys corresponding to sets of attributes with the added restriction that none of these satisfies $(\mathbf{A}, \rho, \mathcal{T}_0)$ and $(\mathbf{A}, \rho, \mathcal{T}_1)$.

Guess The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0, 1\}$ for β and wins the game if $\beta = \beta'$.

The advantage of the adversary in this game is defined as $|\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|$ where the probability is taken over the random bits used by the challenger and the adversary.

DEFINITION 7. *The access structures of a ciphertext-policy attribute-based encryption scheme is partially hidden if all polynomial time adversaries have at most a negligible advantage in this security game.*

3.2 Our Construction

The proposed CP-ABE scheme consists of the following algorithms:

$\mathsf{Setup}(1^\lambda, U)$ The setup algorithm first runs $\mathcal{G}(1^\lambda)$ to obtain $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$ with $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1 p_2 p_3 p_4$. The attribute universe description $U = \mathbb{Z}_N$. Next it chooses $g, h, u_1, \ldots, u_n \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$,$X_4, Z \in \mathbb{G}_{p_4}$ and $\alpha, a \in \mathbb{Z}_N$ uniformly at random. The public parameters are published as

$$\mathsf{PK} = (N, g, g^a, e(g, g)^\alpha, u_1, \ldots, u_n, H = h \cdot Z, X_4).$$

The master secret key is $\mathsf{MSK} = (h, X_3, \alpha)$.

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathcal{S} = (s_1, \ldots, s_n))$ The key generation algorithm chooses $t \in \mathbb{Z}_N$ and $R, R', R_1, \ldots, R_n \in \mathbb{G}_{p_3}$ uniformly at random. The secret key $\mathsf{SK}_{\mathcal{S}} = (\mathcal{S},\ K, K', \{K_i\}_{1 \leq i \leq n})$ is computed as

$$K = g^\alpha g^{at} R,\ K' = g^t R',\ K_i = (u_i^{s_i} h)^t R_i.$$

$\mathsf{Encrypt}(\mathsf{PK}, M \in \mathbb{G}_T, (\mathbf{A}, \rho, \mathcal{T}))$ $\mathbf{A}$ is an $\ell \times n$ matrix, ρ is a map from each row A_x of $\mathbf{A}$ to an attribute name and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$. The encryption algorithm chooses two random vectors $v, v' \in \mathbb{Z}_N^n$, denoted $v = (s, v_2, \ldots, v_n)$ and $v' = (s', v_2', \ldots, v_n')$. It also chooses $r_x, r_x' \in \mathbb{Z}_N$ and $Z_{1,x}, Z_{1,x}', Z_{2,x}, Z_{2,x}' \in \mathbb{G}_{p_4}$ uniformly at random, for $1 \leq x \leq \ell$. The ciphertext is $C = ((\mathbf{A}, \rho),\ \tilde{C}_1, C_1', \{C_{1,x}, D_{1,x}\}_{1 \leq x \leq \ell},\ \tilde{C}_2, C_2', \{C_{2,x}, D_{2,x}\}_{1 \leq x \leq \ell})$, where

$$\tilde{C}_1 = M \cdot e(g, g)^{\alpha s},\ C_1' = g^s,$$

$$C_{1,x} = g^{aA_x \cdot v} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x} \cdot Z_{1,x},\ D_{1,x} = g^{r_x} \cdot Z_{1,x}',$$

$$\tilde{C}_2 = e(g, g)^{\alpha s'},\ C_2' = g^{s'},$$

$$C_{2,x} = g^{aA_x \cdot v'} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x'} \cdot Z_{2,x},\ D_{2,x} = g^{r_x'} \cdot Z_{2,x}'.$$

$\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_{\mathcal{S}}, C)$ Let $C = ((\mathbf{A}, \rho),\ \tilde{C}_1,\ C_1',\ \{C_{1,x}, D_{1,x}\}_{1 \leq x \leq \ell},\ \tilde{C}_2, C_2', \{C_{2,x}, D_{2,x}\}_{1 \leq x \leq \ell})$, $\mathsf{SK}_{\mathcal{S}} = (\mathcal{S}, K,\ K', \{K_i\}_{1 \leq i \leq n})$ and $\mathcal{S} = (s_1, \ldots, s_n)$. The decryption algorithm first calculates $\mathrm{I}_{\mathbf{A},\rho}$ from $(\mathbf{A}, \rho)$, where $\mathrm{I}_{\mathbf{A},\rho}$ denotes the set of minimum subsets of $\{1, \ldots, \ell\}$ that satisfies $(\mathbf{A}, \rho)$. It then checks if there exists an $\mathcal{I} \in \mathrm{I}_{\mathbf{A},\rho}$ that satisfies

$$\tilde{C}_2 = e(C_2', K) / \left(\prod_{i \in \mathcal{I}} (e(C_{2,i}, K') \cdot e(D_{2,i}, K_{\rho(i)}))^{\omega_i} \right),$$

where $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$. If no element in $\mathrm{I}_{\mathbf{A},\rho}$ satisfies the above equation, it outputs $\perp$. Otherwise, it computes

$$\begin{aligned} & e(C_1', K) / \left(\prod_{i \in \mathcal{I}} (e(C_{1,i}, K') \cdot e(D_{1,i}, K_{\rho(i)}))^{\omega_i} \right) \\ & = e(g, g)^{\alpha s} e(g, g)^{ats} / \left(\prod_{i \in \mathcal{I}} e(g, g)^{atA_i \cdot v \cdot \omega_i} \right) \\ & = e(g, g)^{\alpha s}. \end{aligned}$$

Then M can be recovered as $\tilde{C}_1 / e(g, g)^{\alpha s}$.

In our construction, a ciphertext includes two parts: $(\tilde{C}_1,\ C_1',\ \{C_{1,x},\ D_{1,x}\}_{1 \leq x \leq \ell})$ and $(\tilde{C}_2,\ C_2',\ \{C_{2,x},\ D_{2,x}\}_{1 \leq x \leq \ell})$. The first part is an encryption of the message M. The second part is redundant, and can be viewed as an encryption of 1. If the private key attributes of a user satisfy the access structure associated with the ciphertext, the redundant second part will help the user decide which attribute set satisfies the access structure; and then the user is able to use the information and his private key to decrypt the first part of the ciphertext and recover the plaintext M. The CP-ABE construction in [14] uses composite order bilinear groups whose order is the product of three distinct primes, while our construction uses groups whose order is the product of four distinct primes. Note that in our construction, component H of the public parameters and components $C_{1,x}, D_{1,x}, C_{2,x}, D_{2,x}$ of the ciphertext all have an element from $\mathbb{G}_{p_4}$ as a factor. This formation of $H, C_{1,x}, D_{1,x}, C_{2,x}, D_{2,x}$ allows us to prove that the access structures of our CP-ABE scheme is partially hidden. We now state the security theorem of our CP-ABE scheme.

THEOREM 1. *If Assumptions 1, 2, 3, and 4 hold, then the access structures of the proposed CP-ABE is partially hidden.*

PROOF. Following the approach by Lewko and Waters [16], we define two additional structures: *semi-functional* ciphertexts and *semi-functional* keys. These will not be used in the real system, but will be used in our proof.

Semi-functional Ciphertext Let g_2 denote a generator of the subgroup $\mathbb{G}_{p_2}$. A semi-functional ciphertext is created as follows. We first use the encryption algorithm to form a normal ciphertext $C' = ((\mathbf{A}, \rho),\ \tilde{C}'_1,\ C''_1,\ \{C'_{1,x},\ D'_{1,x}\}_{1 \leq x \leq \ell},\ \tilde{C}'_2, C''_2, \{C'_{2,x}, D'_{2,x}\}_{1 \leq x \leq \ell})$. Then, we choose random exponents $c, c' \in \mathbb{Z}_N$ and two random vectors $w, w' \in \mathbb{Z}_N^n$. We also choose random values $z_i \in \mathbb{Z}_N$ associated to attributes, and random values $\gamma_x, \gamma'_x \in \mathbb{Z}_N$ associated to row x of the $\ell \times n$ matrix $\mathbf{A}$. The semi-functional ciphertext C is set to be

$$\Big((\mathbf{A}, \rho),\ \tilde{C}_1 = \tilde{C}'_1,\ C'_1 = C''_1 \cdot g_2^{c},$$

$$\{C_{1,x} = C'_{1,x} \cdot g_2^{A_x w + \gamma_x z_{\rho(x)}},$$

$$D_{1,x} = D'_{1,x} \cdot g_2^{-\gamma_x}\}_{1 \leq x \leq \ell},$$

$$C'_2 = C''_2 \cdot g_2^{c'},$$

$$\{C_{2,x} = C'_{2,x} \cdot g_2^{A_x w' + \gamma'_x z_{\rho(x)}},$$

$$D_{2,x} = D'_{2,x} \cdot g_2^{-\gamma'_x}\}_{1 \leq x \leq \ell}\Big).$$

Semi-functional Key A semi-functional key will take on one of three forms. To create a semi-functional key, we first use the key generation algorithm to form a normal key $\mathsf{SK}'_{\mathcal{S}} = (\mathcal{S},\ K', K'', \{K'_i\}_{1 \leq i \leq n})$. Then, we choose random exponents $d, d', d_i \in \mathbb{Z}_N$. The semi-functional key of type 1 is set as

$$(\mathcal{S},\ K = K' {\cdot} g_2^{d},\ K' = K'' {\cdot} g_2^{d'},\ \{K_i = K'_i {\cdot} g_2^{d' z_i}\}_{1 \leq i \leq n}).$$

The semi-functional key of type 2 is set as

$$(\mathcal{S},\ K = K' \cdot g_2^{d},\ K' = K'',\ \{K_i = K'_i\}_{1 \leq i \leq n}).$$

The semi-functional key of type 3 is set as

$$(\mathcal{S},\ K = K' {\cdot} g_2^{d},\ K' = K'' {\cdot} g_2^{d'},\ \{K_i = K'_i {\cdot} g_2^{d_i}\}_{1 \leq i \leq n}).$$

We will prove the security of our scheme from Assumptions 1, 2, 3 and 4 using a hybrid argument over a sequence of games. The first game, Game_{real} is the real security game (the ciphertext and all the keys are normal). In the next game, Game_0, all of the keys will be normal, but the challenge ciphertext will be semi-functional. We let q denote the number of key queries made by the attacker. For k from 1 to q, we define

$\mathsf{Game}_{k,1}$ In this game, the challenge ciphertext is semi- functional, the first $k-1$ keys are semi-functional of type 3, the k^{th} key is semi-functional of type 1, and the remaining keys are normal.

$\mathsf{Game}_{k,2}$ In this game, the challenge ciphertext is semi- functional, the first $k-1$ keys are semi-functional of type 3, the k^{th} key is semi-functional of type 2, and the remaining keys are normal.

$\mathsf{Game}_{k,3}$ In this game, the challenge ciphertext is semi- functional, the first k keys are semi-functional of type 3, and the remaining keys are normal.

For notational purposes, we think of $\mathsf{Game}_{0,3}$ as another way of denoting Game_0. We note that in $\mathsf{Game}_{q,3}$, all of the keys are semi-functional of type 3. In the penultimate game, $\mathsf{Game}_{\mathsf{Final}_0}$, all the keys are semi-functional, and the ciphertext is a semi-functional encryption of a random message, independent of the messages M_0 and M_1 provided by the adversary. The final game, $\mathsf{Game}_{\mathsf{Final}_1}$, is the same as $\mathsf{Game}_{\mathsf{Final}_0}$, except that in the challenge ciphertext, $C_{1,x}$ and $C_{2,x}$ are chosen from $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$ at random (thus the ciphertext is independent from $\mathcal{T}_0$ and $\mathcal{T}_1$ provided by the adversary). It is clear that in the final game, no adversary can have advantage greater than 0.

We prove that these games are indistinguishable in six lemmas, whose formal descriptions and proofs are given in the Appendix. Therefore, we conclude that the advantage of the adversary in Game_{real} (i.e., the real security game) is negligible. This completes the proof of Theorem 1. $\square$

4. CONCLUSIONS

In this paper, we presented an efficient CP-ABE scheme with partially hidden access structures. Our scheme can handle any access structure that can be expressed as an LSSS. Previous CP-ABE schemes with partially hidden access structures [22, 19, 13] only support restricted access structures, which can be expressed as AND gates on multi-valued attributes with wildcards; thus our scheme is more flexible and expressive.

By applying the dual system encryption methodology [29], we proved that our scheme is fully secure in the standard model. The security of our scheme relies on some non-standard complexity assumptions. A further direction is to find expressive CP-ABE constructions with partially hidden access structures from simple assumptions.

5. ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their helpful comments. This work is supported by the Office of Research, Singapore Management University.

6. REFERENCES

[1] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* PhD thesis, Israel Institute of Technology, 1996.

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[3] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[4] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, pages 325–341, 2005.

[5] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[6] A. D. Caro, V. Iovino, and G. Persiano. Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts. In *Pairing*, pages 347–366, 2010.

[7] M. Chase. Multi-authority attribute based encryption. In *TCC*, pages 515–534, 2007.

[8] M. Chase and S. S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM Conference on Computer and Communications Security*, pages 121–130, 2009.

[9] L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. In *ACM Conference on Computer and Communications Security*, pages 456–465, 2007.

[10] V. Goyal, A. J. 0002, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[12] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[13] J. Lai, R. H. Deng, and Y. Li. Fully secure cipertext-policy hiding CP-ABE. In *ISPEC*, pages 24–39, 2011.

[14] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.

[15] A. B. Lewko, Y. Rouselakis, and B. Waters. Achieving leakage resilience through dual system encryption. In *TCC*, pages 70–88, 2011.

[16] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, pages 455–479, 2010.

[17] A. B. Lewko and B. Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, pages 568–588, 2011.

[18] A. B. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In *EUROCRYPT*, pages 547–567, 2011.

[19] J. Li, K. Ren, B. Zhu, and Z. Wan. Privacy-aware attribute-based encryption with user accountability. In *ISC*, pages 347–362, 2009.

[20] H. Lin, Z. Cao, X. Liang, and J. Shao. Secure threshold multi authority attribute based encryption without a central authority. In *INDOCRYPT*, pages 426–436, 2008.

[21] S. Müller, S. Katzenbeisser, and C. Eckert. Distributed attribute-based encryption. In *ICISC*, pages 20–36, 2008.

[22] T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In *ACNS*, pages 111–129, 2008.

[23] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.

[24] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.

[25] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203, 2007.

[26] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[27] E. Shen, E. Shi, and B. Waters. Predicate privacy in encryption systems. In *TCC*, pages 457–473, 2009.

[28] E. Shi and B. Waters. Delegating capabilities in predicate encryption systems. In *ICALP (2)*, pages 560–578, 2008.

[29] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

[30] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography*, pages 53–70, 2011.

APPENDIX

A. SECURITY PROOFS

LEMMA 1. *Suppose that $\mathcal{G}$ satisfies Assumption 1. Then Game_{real} and Game_0 are computationally indistinguishable.*

PROOF. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes Game_{real} and Game_0. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 1. $\mathcal{B}$ is given g, X_3, X_4, T and will simulate Game_{real} or Game_0 with $\mathcal{A}$. $\mathcal{B}$ chooses $\alpha, a, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ and $Z \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $h = g^{a_0}, u_1 = g^{a_1}, \ldots, u_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (N, g, g^a, e(g,g)^\alpha, u_1, \ldots, u_n, H = h \cdot Z, X_4).$$

It can generate normal keys in response to $\mathcal{A}$'s key requests by using the key generation algorithm, since it knows the $\mathsf{MSK} = (h, X_3, \alpha)$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. $\mathcal{B}$ chooses random values $\tilde{v}_2, \ldots, \tilde{v}_n, \tilde{v}'_2, \ldots, \tilde{v}'_n \in \mathbb{Z}_N$ and creates vectors $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$, $\tilde{v}'_n = (1, \tilde{v}'_2, \ldots, \tilde{v}'_n)$.
2. $\mathcal{B}$ chooses random values $\tilde{r}_x, \tilde{r}'_x \in \mathbb{Z}_N$ and $\tilde{Z}_{1,x}, Z'_{1,x}, \tilde{Z}_{2,x}, Z'_{2,x} \in \mathbb{G}_{p_4}$ for $1 \le x \le \ell$.
3. Let $\mathcal{T}_\beta = (t_{\rho(1)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses random exponent $\tilde{s} \in \mathbb{Z}_N$ and computes
$$\begin{aligned} &\tilde{C}_1 = M_\beta \cdot e(g^\alpha, T),\ C'_1 = T, \\ &\qquad C_{1,x} = T^{aA_x \cdot \tilde{v}} \cdot T^{-(a_0 + a_{\rho(x)} t_{\rho(x)})\tilde{r}_x} \cdot \tilde{Z}_{1,x}, \\ &\qquad D_{1,x} = T^{\tilde{r}_x} \cdot Z'_{1,x}, \\ &\tilde{C}_2 = e(g^\alpha, T^{\tilde{s}}),\ C'_2 = T^{\tilde{s}}, \\ &\qquad C_{2,x} = T^{\tilde{s}aA_x \cdot \tilde{v}'} \cdot T^{-(a_0 + a_{\rho(x)} t_{\rho(x)})\tilde{r}'_x} \cdot \tilde{Z}_{2,x}, \\ &\qquad D_{2,x} = T^{\tilde{r}'_x} \cdot Z'_{2,x}. \end{aligned}$$
4. $\mathcal{B}$ sets the challenge ciphertext as $C = ((\mathbf{A}, \rho),\ \tilde{C}_1,\ C'_1,$ $\{C_{1,x}, D_{1,x}\}_{1 \le x \le \ell},\ \tilde{C}_2,\ C'_2,\ \{C_{2,x}, D_{2,x}\}_{1 \le x \le \ell})$ and sends it to $\mathcal{A}$.

If $T \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, let $T = g^s g_2^c$, then

$$\begin{aligned} &\tilde{C}_1 = M_\beta \cdot e(g,g)^{\alpha s},\ C'_1 = g^s \cdot g_2^c, \\ &\qquad C_{1,x} = g^{aA_x \cdot v}(u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x} Z_{1,x} \cdot g_2^{A_x w + \gamma_x z_{\rho(x)}}, \\ &\qquad D_{1,x} = g^{r_x} Z'_{1,x} \cdot g_2^{-\gamma_x}, \\ &\tilde{C}_2 = e(g,g)^{\alpha s'},\ C'_2 = g^{s'} \cdot g_2^{c'}, \\ &\qquad C_{2,x} = g^{aA_x \cdot v'}(u_{\rho(x)}^{t_{\rho(x)}} H)^{-r'_x} Z_{2,x} \cdot g_2^{A_x w' + \gamma'_x z_{\rho(x)}}, \\ &\qquad D_{2,x} = g^{r'_x} Z'_{2,x} \cdot g_2^{-\gamma'_x}, \end{aligned}$$

where $s' = s\tilde{s}$, $c' = c\tilde{s}$, $v = (s, s\tilde{v}_2, \ldots, s\tilde{v}_n)$, $v' = (s',\ s'\tilde{v}'_2, \ldots, s'\tilde{v}'_n)$, $r_x = s\tilde{r}_x$, $r'_x = s\tilde{r}'_x$, $Z_{1,x} = \tilde{Z}_{1,x}Z^{r_x}$, $Z_{2,x} = \tilde{Z}_{2,x}Z^{r'_x}$, $w = ca\tilde{v}$, $w' = c\tilde{s}a\tilde{v}'$, $\gamma_x = -c\tilde{r}_x$, $\gamma'_x = -c\tilde{r}'_x$, $z_{\rho(x)} = a_0 + a_{\rho(x)}t_{\rho(x)}$. This is a semi-functional ciphertext and $\mathcal{B}$ simulates Game_0. We note that the values of $a, a_0, a_{\rho(x)}, t_{\rho(x)}, \tilde{s}, \tilde{v}_2, \ldots, \tilde{v}_n, \tilde{v}'_2, \ldots, \tilde{v}'_n, \tilde{r}_x, \tilde{r}'_x$ modulo p_1 are uncorrelated from their values modulo p_2, so this is properly distributed. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1}$, it is easy to observe that this is a normal ciphertext and $\mathcal{B}$ simulates Game_{real}. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

LEMMA 2. *Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathsf{Game}_{k-1,3}$ and $\mathsf{Game}_{k,1}$ are computationally indistinguishable.*

PROOF. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{k-1,3}$ and $\mathsf{Game}_{k,1}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 2. $\mathcal{B}$ is given $g, X_1X_2, Y_2Y_3, X_3, X_4, T$ and will simulate $\mathsf{Game}_{k-1,3}$ or $\mathsf{Game}_{k,1}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $\alpha, a, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ and $Z \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $h = g^{a_0}, u_1 = g^{a_1}, \ldots, u_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (N, g, g^a, e(g,g)^\alpha, u_1, \ldots, u_n, H = h \cdot Z, X_4).$$

Note that $\mathcal{B}$ knows the master secret key $\mathsf{MSK} = (h, X_3, \alpha)$ associated with PK. Let us now explain how $\mathcal{B}$ answers the j-th key query for $\mathcal{S} = (s_1, \ldots, s_n)$.

For $j < k$, $\mathcal{B}$ creates a semi-functional key of type 3 by choosing random exponents $t, \tilde{d}, \tilde{d}', \tilde{d}'_1, \ldots, \tilde{d}'_n \in \mathbb{Z}_N$, and setting:

$$K = g^\alpha g^{at}(Y_2Y_3)^{\tilde{d}}, \qquad K' = g^t(Y_2Y_3)^{\tilde{d}'},$$
$$\{K_i = (u_i^{s_i}h)^t(Y_2Y_3)^{\tilde{d}'_i}\}_{1\le i\le n}.$$

We note that this is a properly distributed semi-functional key of 3 because the values of $\tilde{d}, \tilde{d}', \tilde{d}'_i$ modulo p_2 are uncorrelated to their values modulo p_3.

For $j > k$, $\mathcal{B}$ creates a normal key by running the key generation algorithm since it knows the MSK.

To answer the k-th key quest for $\mathcal{S} = (s_1, \ldots, s_n)$, $\mathcal{B}$ chooses random elements $\tilde{R}, \tilde{R}', \tilde{R}_1, \ldots, \tilde{R}_n \in \mathbb{G}_{p_3}$ and sets:

$$K = g^\alpha \cdot T^a \cdot \tilde{R},\ K' = T \cdot \tilde{R}',\ \{K_i = T^{a_0 + a_i s_i} \cdot \tilde{R}'_i\}_{i\le i\le n}.$$

We have the following observations. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then T can be written as $g^t g_2^{d'} \bar{R}$, and

$$K = g^\alpha g^{at} R \cdot g_2^d, \qquad K' = g^t R' \cdot g_2^{d'},$$
$$\{K_i = (u_i^{s_i}h)^t R_i \cdot g_2^{d' z_i}\}_{i\le i\le n},$$

where $R = \bar{R}^a\tilde{R}, d = ad', R' = \bar{R}\tilde{R}', R_i = \bar{R}^{a_0+a_is_i}\tilde{R}'_i, z_i = a_0 + a_is_i$. This is a semi-function key of type 1. Note that the values of a, a_0, a_i, s_i modulo p_1 are uncorrelated from their values modulo p_2. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed normal key.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. $\mathcal{B}$ chooses random values $\tilde{v}_2, \ldots, \tilde{v}_n, \tilde{v}'_2, \ldots, \tilde{v}'_n \in \mathbb{Z}_N$ and creates vectors $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$, $\tilde{v}'_n = (1, \tilde{v}'_2, \ldots, \tilde{v}'_n)$.
2. $\mathcal{B}$ chooses random values $\tilde{r}_x, \tilde{r}'_x \in \mathbb{Z}_N$ and $\tilde{Z}_{1,x}, Z'_{1,x}, \tilde{Z}_{2,x}, Z'_{2,x} \in \mathbb{G}_{p_4}$ for $1 \le x \le \ell$.
3. Let $\mathcal{T}_\beta = (t_{\rho(1)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses random exponent $\tilde{s} \in \mathbb{Z}_N$ and computes
$$\tilde{C}_1 = M_\beta \cdot e(g^\alpha, X_1X_2),\ C'_1 = X_1X_2,$$
$$C_{1,x} = (X_1X_2)^{aA_x\cdot\tilde{v}} \cdot (X_1X_2)^{-(a_0+a_{\rho(x)}t_{\rho(x)})\tilde{r}_x} \cdot \tilde{Z}_{1,x},$$
$$D_{1,x} = (X_1X_2)^{\tilde{r}_x} \cdot Z'_{1,x},$$
$$\tilde{C}_2 = e(g^\alpha, (X_1X_2)^{\tilde{s}}),\ C'_2 = (X_1X_2)^{\tilde{s}},$$
$$C_{2,x} = (X_1X_2)^{\tilde{s}aA_x\cdot\tilde{v}'} \cdot (X_1X_2)^{-(a_0+a_{\rho(x)}t_{\rho(x)})\tilde{r}'_x} \cdot \tilde{Z}_{2,x},$$
$$D_{2,x} = (X_1X_2)^{\tilde{r}'_x} \cdot Z'_{2,x}.$$
4. $\mathcal{B}$ sets the challenge ciphertext as $C = ((\mathbf{A}, \rho),\ \tilde{C}_1,\ C'_1,\ \{C_{1,x}, D_{1,x}\}_{1\le x\le\ell},\ \tilde{C}_2, C'_2, \{C_{2,x}, D_{2,x}\}_{1\le x\le\ell})$ and sends it to $\mathcal{A}$.

If we let $X_1X_2 = g^s g_2^c$, then

$$\tilde{C}_1 = M_\beta \cdot e(g,g)^{\alpha s},\ C'_1 = g^s \cdot g_2^c,$$
$$C_{1,x} = g^{aA_x\cdot v}(u_{\rho(x)}^{t_{\rho(x)}}H)^{-r_x}Z_{1,x} \cdot g_2^{A_xw+\gamma_x z_{\rho(x)}},$$
$$D_{1,x} = g^{r_x}Z'_{1,x} \cdot g_2^{-\gamma_x},$$
$$\tilde{C}_2 = e(g,g)^{\alpha s'},\ C'_2 = g^{s'} \cdot g_2^{c'},$$
$$C_{2,x} = g^{aA_x\cdot v'}(u_{\rho(x)}^{t_{\rho(x)}}H)^{-r'_x}Z_{2,x} \cdot g_2^{A_xw'+\gamma'_x z_{\rho(x)}},$$
$$D_{2,x} = g^{r'_x}Z'_{2,x} \cdot g_2^{-\gamma'_x},$$

where $s' = s\tilde{s}$, $c' = c\tilde{s}$, $v = (s, s\tilde{v}_2, \ldots, s\tilde{v}_n)$, $v' = (s', s'\tilde{v}'_2, \ldots, s'\tilde{v}'_n)$, $r_x = s\tilde{r}_x$, $r'_x = s\tilde{r}'_x$, $Z_{1,x} = \tilde{Z}_{1,x}Z^{r_x}$, $Z_{2,x} = \tilde{Z}_{2,x}Z^{r'_x}$, $w = ca\tilde{v}$, $w' = c\tilde{s}a\tilde{v}'$, $\gamma_x = -c\tilde{r}_x$, $\gamma'_x = -c\tilde{r}'_x$, $z_{\rho(x)} = a_0 + a_{\rho(x)}t_{\rho(x)}$. This is a semi-functional ciphertext. Note that the values of $a,\ a_0,\ a_{\rho(x)},\ t_{\rho(x)},\ \tilde{s},\ \tilde{v}_2,\ \ldots,\ \tilde{v}_n, \tilde{v}'_2, \ldots, \tilde{v}'_n, \tilde{r}_x, \tilde{r}'_x$ modulo p_1 are uncorrelated from their values modulo p_2.

Similar to the analysis in the proof of Lemma 2 of Lewko et al.'s CP-ABE scheme [14], the k^{th} key and ciphertext are properly distributed. We can thus conclude that, if $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,1}$. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k-1,3}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

LEMMA 3. *Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathsf{Game}_{k,1}$ and $\mathsf{Game}_{k,2}$ are computationally indistinguishable.*

PROOF. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{k,1}$ and $\mathsf{Game}_{k,2}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 2. $\mathcal{B}$ is given $g, X_1X_2, Y_2Y_3, X_3, X_4, T$ and will simulate $\mathsf{Game}_{k,1}$ or $\mathsf{Game}_{k,2}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $\alpha, a, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ and $Z \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $h = g^{a_0}, u_1 = g^{a_1}, \ldots, u_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (N, g, g^a, e(g,g)^\alpha, u_1, \ldots, u_n, H = h \cdot Z, X_4).$$

The first $k-1$ semi-functional keys of type 3, the normal keys $> k$, and the challenge ciphertext are constructed exactly as in the Lemma 2.

To answer the k-th key quest for $\mathcal{S} = (s_1, \ldots, s_n)$, $\mathcal{B}$ proceeds as it did in the Lemma 2, but $\mathcal{B}$ additionally chooses a random exponent $\delta \in \mathbb{Z}_N$ and sets:

$$K = g^\alpha \cdot T^a \cdot \tilde{R} \cdot (Y_2Y_3)^\delta, \qquad K' = T \cdot \tilde{R}',$$
$$\{K_i = T^{a_0+a_is_i} \cdot \tilde{R}'_i\}_{i\le i\le n},$$

The only change we have made here is adding the $(Y_2Y_3)^\delta$ term, which randomizes the $\mathbb{G}_{p_2}$ part of K. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 1. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 2.

We can conclude that, if $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,1}$. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,2}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

LEMMA 4. *Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathsf{Game}_{k,2}$ and $\mathsf{Game}_{k,3}$ are computationally indistinguishable.*

PROOF. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{k,2}$ and $\mathsf{Game}_{k,3}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 2. $\mathcal{B}$ is given $g, X_1X_2, Y_2Y_3, X_3, X_4, T$ and will simulate $\mathsf{Game}_{k,2}$ or $\mathsf{Game}_{k,3}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $\alpha, a, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ and $Z \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $h = g^{a_0}, u_1 = g^{a_1}, \ldots, u_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (N, g, g^a, e(g,g)^\alpha, u_1, \ldots, u_n, H = h \cdot Z, X_4).$$

The first $k-1$ semi-functional keys of type 3, the normal keys $> k$, and the challenge ciphertext are constructed exactly as in the Lemma 2.

To answer the k-th key quest for $\mathcal{S} = (s_1, \ldots, s_n)$, $\mathcal{B}$ chooses a random exponent $\delta \in \mathbb{Z}_N$, random elements $\tilde{R}$, $\tilde{R}'$, $\tilde{R}_1, \ldots, \tilde{R}_n \in \mathbb{G}_{p_3}$ and sets:

$$K = g^\alpha \cdot T^a \cdot \tilde{R} \cdot (Y_2Y_3)^\delta, \qquad K' = T \cdot \tilde{R}',$$
$$\{K_i = T^{a_0 + a_i s_i} \cdot \tilde{R}'_i\}_{i \le i \le n}.$$

We have the following observations. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then T can be written as $g^t g_2^{d'} \bar{R}$, and

$$K = g^\alpha g^{at} R \cdot g_2^d, \qquad K' = g^t R' \cdot g_2^{d'},$$
$$\{K_i = (u_i^{s_i} h)^t R_i \cdot g_2^{d_i}\}_{i \le i \le n},$$

where $R = \bar{R}^a \tilde{R} Y_3^\delta$, $g_2^d = g_2^{ad'} Y_2^\delta$, $R' = \bar{R}\tilde{R}'$, $R_i = \bar{R}^{a_0 + a_i s_i} \tilde{R}'_i$, $d_i = d'(a_0 + a_i s_i)$. This is a semi-functional key of type 3. Note that the values of δ modulo p_2 are uncorrelated from their values modulo p_3. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 2.

Similar to the analysis in the proof of Lemma 2 of Lewko et al.'s CP-ABE scheme [14], the k^{th} key and ciphertext are properly distributed. We can conclude that, if $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,3}$. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,2}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

LEMMA 5. *Suppose that $\mathcal{G}$ satisfies Assumption 3. Then $\mathsf{Game}_{q,3}$ and $\mathsf{Game}_{\mathsf{Final}_0}$ are computationally indistinguishable.*

PROOF. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{q,3}$ and $\mathsf{Game}_{\mathsf{Final}_0}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 3. $\mathcal{B}$ is given $g, g_2, g^\alpha X_2, g^s Y_2, X_3, X_4, T$ and will simulate $\mathsf{Game}_{q,3}$ or $\mathsf{Game}_{\mathsf{Final}_0}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $a, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ and $Z \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $h = g^{a_0}, u_1 = g^{a_1}, \ldots, u_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (N, g, g^a,\ e(g, g^\alpha X_2) = e(g,g)^\alpha,$$
$$u_1, \ldots, u_n, H = h \cdot Z, X_4).$$

Each time $\mathcal{B}$ is asked to provide a key for $\mathcal{S} = (s_1, \ldots, s_n)$, $\mathcal{B}$ creates a semi-functional key of type 3 by choosing random exponents $t, \tilde{d}, d', d_1, \ldots, d_n \in \mathbb{Z}_N$, random elements $R, R', R_1, \ldots, R_n \in \mathbb{G}_{p_3}$, and setting:

$$K = (g^\alpha X_2) g^{at} R \cdot g_2^{\tilde{d}}, \qquad K' = g^t R' \cdot g_2^{d'},$$
$$\{K_i = (u_i^{s_i} h)^t R_i \cdot g_2^{d_i}\}_{1 \le i \le n}.$$

We note that K can be written as $g^\alpha g^{at} R \cdot g_2^d$, where $g_2^d = X_2 g_2^{\tilde{d}}$, so this is a properly distributed semi-functional key of type 3.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. $\mathcal{B}$ chooses random values $\tilde{v}_2, \ldots, \tilde{v}_n \in \mathbb{Z}_N$ and creates the vector $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$. $\mathcal{B}$ also chooses two random vectors $v' = (s', v'_2, \ldots, v'_n), w' = (w'_1, \ldots, w'_n) \in \mathbb{Z}_N^n$.
2. $\mathcal{B}$ chooses random values $\tilde{r}_x, r'_x, \gamma'_x \in \mathbb{Z}_N$ and $\tilde{Z}_{1,x}, Z'_{1,x}, Z_{2,x}, Z'_{2,x} \in \mathbb{G}_{p_4}$ for $1 \le x \le \ell$.
3. Let $\mathcal{T}_\beta = (t_{\rho(1)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses random exponent $c' \in \mathbb{Z}_N$ and computes
$$\tilde{C}_1 = M_\beta \cdot T,\ C'_1 = g^s Y_2,$$
$$C_{1,x} = (g^s Y_2)^{aA_x \cdot \tilde{v}} (g^s Y_2)^{-(a_0 + a_{\rho(x)} t_{\rho(x)})\tilde{r}_x} \tilde{Z}_{1,x},$$
$$D_{1,x} = (g^s Y_2)^{\tilde{r}_x} \cdot Z'_{1,x},$$
$$\tilde{C}_2 = e(g,g)^{\alpha s'},\ C'_2 = g^{s'} g_2^{c'},$$
$$C_{2,x} = g^{aA_x \cdot v'} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r'_x} Z_{2,x} g_2^{A_x w' + \gamma'_x (a_0 + a_{\rho(x)} t_{\rho(x)})},$$
$$D_{2,x} = g^{r'_x} Z'_{2,x} \cdot g_2^{-\gamma'_x}.$$
4. $\mathcal{B}$ sets the challenge ciphertext as $C = ((\mathbf{A}, \rho),\ \tilde{C}_1,\ C'_1, \{C_{1,x}, D_{1,x}\}_{1 \le x \le \ell},\ \tilde{C}_2, C'_2, \{C_{2,x}, D_{2,x}\}_{1 \le x \le \ell})$ and sends it to $\mathcal{A}$.

Let $g^s Y_2 = g^s g_2^c$, then

$$\tilde{C}_1 = M_\beta \cdot T,\ C'_1 = g^s \cdot g_2^c,$$
$$C_{1,x} = g^{aA_x \cdot v} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x} Z_{1,x} \cdot g_2^{A_x w + \gamma_x z_{\rho(x)}},$$
$$D_{1,x} = g^{r_x} Z'_{1,x} \cdot g_2^{-\gamma_x},$$
$$\tilde{C}_2 = e(g,g)^{\alpha s'},\ C'_2 = g^{s'} \cdot g_2^{c'},$$
$$C_{2,x} = g^{aA_x \cdot v'} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r'_x} Z_{2,x} \cdot g_2^{A_x w' + \gamma'_x z_{\rho(x)}},$$
$$D_{2,x} = g^{r'_x} Z'_{2,x} \cdot g_2^{-\gamma'_x},$$

where $v = (s, s\tilde{v}_2, \ldots, s\tilde{v}_n), r_x = s\tilde{r}_x, Z_{1,x} = \tilde{Z}_{1,x} Z^{r_x}$, $w = ca\tilde{v}, \gamma_x = -c\tilde{r}_x, z_{\rho(x)} = a_0 + a_{\rho(x)} t_{\rho(x)}$. Note that the values of $a, a_0, a_{\rho(x)}, t_{\rho(x)}, \tilde{v}_2, \ldots, \tilde{v}_n, \tilde{r}_x$ modulo p_1 are uncorrelated from their values modulo p_2.

If $T = e(g,g)^{\alpha s}$, this is a properly distributed semi-functional encryption of M_β and $\mathcal{B}$ simulates $\mathsf{Game}_{q,3}$. Otherwise, this is a properly distributed semi-functional encryption of a random message in $\mathbb{G}_T$ and $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{Final}_0}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

LEMMA 6. *Suppose that $\mathcal{G}$ satisfies Assumption 4. Then* $\mathsf{Game}_{\mathsf{Final}_0}$ *and* $\mathsf{Game}_{\mathsf{Final}_1}$ *are computationally indistinguishable.*

PROOF. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{Final}_0}$ and $\mathsf{Game}_{\mathsf{Final}_1}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 4. $\mathcal{B}$ is given $(g, g_2, g^{t'}B_2, h^{t'}Y_2, X_3, X_4, hZ, g^{r'}D_2D_4, T)$ and will simulate $\mathsf{Game}_{\mathsf{Final}_0}$ or $\mathsf{Game}_{\mathsf{Final}_1}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $a, \alpha, a_1, \ldots, a_n \in \mathbb{Z}_N$ and $Z \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u_1 = g^{a_1}, \ldots, u_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (N, g, g^a, e(g,g)^\alpha, u_1, \ldots, u_n, H = hZ, X_4).$$

Each time $\mathcal{B}$ is asked to provide a key for $\mathcal{S} = (s_1, \ldots, s_n)$, $\mathcal{B}$ creates a semi-functional key by choosing a random exponent $\tilde{t} \in \mathbb{Z}_N$, random elements $R, R', R_1, \ldots, R_n \in \mathbb{G}_{p_3}$, and setting:

$$K = g^\alpha (g^{t'}B_2)^{a\tilde{t}} R, \qquad K' = (g^{t'}B_2)^{\tilde{t}} R',$$
$$\{K_i = (g^{t'}B_2)^{a_i s_i \tilde{t}} (h^{t'}Y_2)^{\tilde{t}} R_i\}_{1 \le i \le n}.$$

We observe that

$$K = g^\alpha g^{at} R \cdot g_2^d, \qquad K' = g^t R' \cdot g_2^{d'},$$
$$\{K_i = (u_i^{s_i} h)^t R_i \cdot g_2^{d_i}\}_{1 \le i \le n},$$

where $t = t'\tilde{t}, g_2^d = B_2^{a\tilde{t}}, g_2^{d'} = B_2^{\tilde{t}}, g_2^{d_i} = B_2^{a_i s_i \tilde{t}} Y_2^{\tilde{t}}$. This is a properly distributed semi-functional key of type 3 because the values of $\tilde{t}, a, a_i, s_i$ modulo p_2 is uncorrelated to their values modulo p_1.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. $\mathcal{B}$ chooses random vectors $v = (s, v_2, \ldots, v_n)$, $v' = (s', v'_2, \ldots, v'_n)$, $w, w' \in \mathbb{Z}_N^n$.
2. $\mathcal{B}$ chooses random values $\tilde{r}_x, \tilde{r}'_x \in \mathbb{Z}_N$ and $\tilde{Z}_{1,x}, \tilde{Z}_{2,x} \in \mathbb{G}_{p_4}$ for $1 \le x \le \ell$.
3. Let $\mathcal{T}_\beta = (t_{\rho(1)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses random exponents $c, c' \in \mathbb{Z}_N$ and sets

$$\tilde{C}_1 \xleftarrow{\$} \mathbb{G}_T,\ C'_1 = g^s g_2^c,$$
$$C_{1,x} = g^{aA_x \cdot v} (g^{r'}D_2D_4)^{-\tilde{r}_x a_{\rho(x)} t_{\rho(x)}} T^{-\tilde{r}_x} g_2^{A_x w} \tilde{Z}_{1,x},$$
$$D_{1,x} = (g^{r'}D_2D_4)^{\tilde{r}_x},$$
$$\tilde{C}_2 = e(g,g)^{\alpha s'},\ C'_2 = g^{s'} g_2^{c'},$$
$$C_{2,x} = g^{aA_x \cdot v'} (g^{r'}D_2D_4)^{-\tilde{r}'_x a_{\rho(x)} t_{\rho(x)}} T^{-\tilde{r}'_x} g_2^{A_x w'} \tilde{Z}_{2,x},$$
$$D_{2,x} = (g^{r'}D_2D_4)^{\tilde{r}'_x}.$$

4. $\mathcal{B}$ sets the challenge ciphertext as $C = ((\mathbf{A}, \rho),\ \tilde{C}_1,\ C'_1,$ $\{C_{1,x}, D_{1,x}\}_{1 \le x \le \ell},\ \tilde{C}_2, C'_2, \{C_{2,x}, D_{2,x}\}_{1 \le x \le \ell})$ and sends it to $\mathcal{A}$.

If $T = h^{r'}A_2A_4$, let $D_2 = g_2^\gamma$ and $A_2 = g_2^{\gamma\delta}$, we have

$$\tilde{C}_1 \xleftarrow{\$} \mathbb{G}_T,\ C'_1 = g^s \cdot g_2^c,$$
$$C_{1,x} = g^{aA_x \cdot v} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x} Z_{1,x} \cdot g_2^{A_x w + \gamma_x z_{\rho(x)}},$$
$$D_{1,x} = g^{r_x} Z'_{1,x} \cdot g_2^{-\gamma_x},$$
$$\tilde{C}_2 = e(g,g)^{\alpha s'},\ C'_2 = g^{s'} \cdot g_2^{c'},$$
$$C_{2,x} = g^{aA_x \cdot v'} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r'_x} Z_{2,x} \cdot g_2^{A_x w' + \gamma'_x z_{\rho(x)}},$$
$$D_{2,x} = g^{r'_x} Z'_{2,x} \cdot g_2^{-\gamma'_x},$$

where $r_x = r'\tilde{r}_x,\ Z_{1,x} = Z^{r_x} \tilde{Z}_{1,x} A_4^{-\tilde{r}_x} D_4^{-\tilde{r}_x a_{\rho(x)} t_{\rho(x)}},\ \gamma_x = -\gamma\tilde{r}_x,\ z_{\rho(x)} = \delta + a_{\rho(x)} t_{\rho(x)},\ Z'_{1,x} = D_4^{\tilde{r}_x},\ Z_{2,x} = Z^{r'_x} \tilde{Z}_{2,x} A_4^{-\tilde{r}'_x} D_4^{-\tilde{r}'_x a_{\rho(x)} t_{\rho(x)}},\ \gamma'_x = -\gamma\tilde{r}'_x,\ Z'_{2,x} = D_4^{\tilde{r}'_x}$. This is a properly distributed semi-functional encryption of a random message in $\mathbb{G}_T$ because the values of $\tilde{r}_x, \tilde{r}'_x, a_{\rho(x)}, t_{\rho(x)}$ modulo p_1 and p_2 are uncorrelated from their values modulo p_4. If $T \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, this is a properly distributed semi-functional ciphertext with $\tilde{C}_1$ random in $\mathbb{G}_T$, and $C_{1,x}, C_{2,x}$ random in $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$.

We can conclude that, if $T = h^{r'}A_2A_4$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Final}_0}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Final}_1}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □