Fully Secure Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts and Fast Decryption

Junzuo Lai
Dept. of Computer Science
Jinan University, China
The State Key Laboratory of
Integrated Services Networks
Xidian University, China
laijunzuo@gmail.com

Robert H. Deng
School of Information Systems
Singapore Management
University
Singapore 178902
robertdeng@smu.edu.sg

Yingjiu Li
School of Information Systems
Singapore Management
University
Singapore 178902
yjli@smu.edu.sg

Jian Weng
Dept. of Computer Science
Jinan University
Guangzhou 510632, China
cryptjweng@gmail.com

ABSTRACT

Attribute-based encryption (ABE), introduced by Sahai and Waters, is a promising cryptographic primitive, which has been widely applied to implement fine-grained access control system for encrypted data. In its key-policy flavor, attribute sets are used to annotate ciphertexts and secret keys are associated with access structures that specify which ciphertexts a user is entitled to decrypt. In most existing key-policy attribute-based encryption (KP-ABE) constructions, the size of the ciphertext is proportional to the number of attributes associated with it and the decryption cost is proportional to the number of attributes used during decryption.

In this paper, we present a new construction of KP-ABE. Our proposed construction is the first KP-ABE scheme, which has the following features simultaneously: expressive (i.e., supporting arbitrary monotonic access structures); fully secure in the standard model; constant-size ciphertexts and fast decryption. The downside of our construction is that secret keys have quadratic size in the number of attributes.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public Key Cryptosystems; D.4.6 [**Security and Protection**]: Access controls, Cryptographic controls

General Terms

Security, Design

ASIA CCS'14, June 4–6, 2014, Kyoto, Japan.

http://dx.doi.org/10.1145/2590296.2590334.

Keywords

Key-Policy Attribute-Based Encryption; Full Security; Constant-Size Ciphertexts; Fast Decryption

1. INTRODUCTION

In the traditional public key encryption or identity-based encryption schemes [35, 10], encrypted data is targeted for decryption by a single known user; hence, they lack the expressiveness needed for more advanced data sharing. Many distributed applications require complex access control mechanisms for encrypted data which is stored in untrusted servers, such as in the cloud. Sahai and Waters [34] addressed this issue by introducing the concept of attribute-based encryption (ABE). ABE enables public key 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: key-policy ABE (KP-ABE) [21] and ciphertext-policy ABE (CP-ABE) [7]. In this paper, our concern is on the former.

In a KP-ABE scheme [21], every ciphertext is associated with a set of attributes, and every user's secret key is associated with an access structure on attributes. A user will be able to decrypt a ciphertext only if the access structure associated with the user's secret key is satisfied by the set of attributes associated with the ciphertext. This access control functionality can be very powerful, but also costly. In most existing KP-ABE constructions, the size of the ciphertext is proportional to the number of attributes associated with it and the decryption cost is proportional to the number of attributes used during decryption. Specifically, it usually requires one pairing operation per attribute used during decryption. To the best of our knowledge, the schemes in [4, 23] are the only efforts to design KP-ABE schemes with constant-size ciphertexts[1] and/or fast decryption.

In [4], Attrapadung et al. first showed that a certain class of identity-based broadcast encryption (IBBE) schemes

[1]Constant-size ciphertexts mean that the size of a ciphertext only depends on the security parameter and not on the number of attributes associated with the ciphertext.

readily yields KP-ABE schemes with monotonic (though LSSS-realizable) access structures via a generic transformation. Then, they presented an IBBE scheme with constant-size ciphertexts, which can be seen as an instance of the function encryption for zero inner-product with constant-size ciphertexts proposed in [2, 3] and is implied by spatial encryption of [12], thus yielding a KP-ABE scheme with constant-size ciphertexts. Their scheme also reduces the number of pairing evaluations to a constant during decryption. However, the generic transformation proposed in [4] from a certain class of IBBE to KP-ABE only guarantees the resulting KP-ABE scheme to be selectively secure (i. e., the adversaries have to make up their mind about their target before seeing the public parameters), which is a weak security model analogous to the selective-ID model [8, 13] in IBE schemes. Moreover, the scheme proposed in [4] brought in an inner-product instance as a basic building block which fundamentally demands a bound on the maximum number of attributes that can appear in a ciphertext.

More recently, based on the KP-ABE scheme (denoted as GPSW scheme) by Goyal et al. [21], Hohenberger and Waters [23] presented a KP-ABE scheme in which a ciphertext can be decrypted with only 2 pairings by increasing the secret key size by a factor of $|\Lambda|$, where Λ is the set of distinct attributes that appear in the secret key. They also presented a generalized construction that allows each user to independently tune various efficiency tradeoffs to their liking on a spectrum where the extremes are GPSW scheme on one end and their very fast scheme on the other. Compared with the scheme proposed in [4], their schemes do not place a limit on the number of attributes used in a ciphertext and have shorter secret key size. However, the size of the ciphertext in [23] is proportional to the number of attributes associated with it. On the other hand, the base construction of KP-ABE scheme with fast decryption proposed in [23] only supports a small universe U of attributes, where $|U|$ is a polynomial in the security parameter. Hohenberger and Waters [23] described how to alter their base construction to accommodate a large universe $U = \{0,1\}^*$ of attributes, but relying on the random oracle (RO) model [6].

1.1 Our Contribution

We first observe that, each attribute can be divided into two parts: an attribute name and its value. In most cases, the total number of attribute names in a system is *polynomial*, and the possible number of values of an attribute name is *exponential*. We use the application scenario in [23] to illustrate this. In a KP-ABE system, an encrypted email can be tagged with a set of attributes, such as "from: Alice", "to: IACR board", "subject: voting", "date: October 1, 2013". The master authority for the system issues secret decryption keys associated with access structures to users, such as giving to Bob a decryption key that enables him to decrypt any ciphertexts that satisfy the following access structure,

"to: Bob" OR ("to: IACR board" AND
(January 1, 2013 $\leq$ "date" $\leq$ December 31, 2013)).

In this application scenario, there only exist several attribute names (i.e., "from", "to", "subject" and "date"), but the possible number of values of an attribute name, such as "date", could be exponential.

Based on the above observation, drawing on the hierarchical identity-based encryption (HIBE) scheme with constant-size ciphertexts by Boneh et al. [9] and the fully secure KP-ABE scheme by Lewko et al. [25], we present in this paper the first KP-ABE scheme which has the following features simultaneously: expressive (i.e., supporting arbitrary monotonic access structure); fully secure (cf. selectively secure) in the standard model; constant-size ciphertexts and fast decryption. The downside of our construction is that secret keys have quadratic size, which comprise $O(\ell \cdot n)$ elements, where n is the total number of attribute names in the system and ℓ is the number of leaf nodes in an access tree/structure. A comparison of our scheme to the schemes in [4, 23] focusing on designing KP-ABE schemes with constant-size ciphertexts and/or fast decryption is given in Table 1. Note that, we take the KP-ABE scheme for a small universe of attributes with the fastest decryption algorithm proposed by Hohenberger and Waters [23] for comparison, since their KP-ABE scheme for a large universe of attributes relies on the random oracle model.

1.2 Related Work

The notion of ABE was introduced by Sahai and Waters as an application of their fuzzy identity-based encryption (IBE) scheme [34], where both ciphertexts and secrete keys are associated with sets of attributes. The decryption of a ciphertext is enabled if and only if the set of attributes for the ciphertext and the set of attributes for the secret key overlap by at least a fixed threshold value d. Goyal et al. [21] formulated two complementary forms of ABE: KP-ABE and CP-ABE. In a KP-ABE scheme, decryption keys are associated with access structures and ciphertexts are associated with sets of attributes. In a CP-ABE scheme, the situation is reversed: decryption keys are associated with sets of attributes while ciphertexts are associated with access structures. There exists a general method to transform KP-ABE to CP-ABE [20].

In terms of the expressive power of access structures, Goyal et al. [21] presented the first KP-ABE supporting monotonic access structures. To enable more flexible access control policy, Ostrovsky et al. [32] presented a KP-ABE system that supports the expression of non-monotonic formulas in key policies. Lewko et al. [25] proposed the first fully secure KP-ABE scheme supporting arbitrary monotonic access formulas. Previous constructions of KP-ABE [21, 32] were only proven to be selectively secure. Lewko and Waters [28] 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. Rouselakis and Waters [33] improved the efficiency of the unbounded KP-ABE scheme proposed in [28]. Recently, with the distinct method, Garg et al. [18] and Gorbunov et al. [19] provided the constructions of KP-ABE for general circuits. The problem of building KP-ABE systems with multiple authorities was investigated in [14, 30, 15]. In virtually all existing KP-ABE schemes, the size of the ciphertext is proportional to the number of attributes associated with it, and the decryption cost is proportional to the number of attributes that have been used for decryption. In this paper, we focus on designing KP-ABE scheme with constant-size ciphertexts and fast decryption. To the best of our knowledge, besides the work of us, there only exist two efforts [4, 23] with the

Scheme	Public parameters size[2]	Private key size[2]	Ciphertext overhead[2]	Decryption cost PAIR.	Decryption cost EXP.	Security
ALP[3] [4]	$O(\bar{n})$	$O(\ell \cdot \bar{n})$	2	2	$O(\|I\| \cdot \|S\|)$	selective
HW [23]	$O(\|U\|)$	$O(\ell \cdot \|\Lambda\|)$	$O(\|S\|)$	2	$O(\|I\| \cdot \|\Delta\|)$	selective[4]
Our scheme	$O(n)$	$O(\ell \cdot n)$	2	2	$O(\|I\| \cdot n)$	full

Table 1: Comparison with other KP-ABE schemes having constant-size ciphertexts and/or fast decryption.

† $\bar{n}$ is the maximum number of attributes that can appear in a ciphertext. ℓ is the number of attributes in an access structure for a key. $|I|$ is the number of attributes used in decryption and $|S|$ denotes the number of attributes associated with the ciphertext. U is the universe of attributes in the system and $|U|$ denotes the number of attributes in U. $|\Lambda|$ denotes the number of distinct attributes that appear in the private key and $|\Delta|$ denotes the number of distinct attributes used in decryption. n is the total number of attribute names/categories in the system, which is normally a small number, as illustrated in the email example.

‡ PAIR. and EXP. denote the number of paring and exponentiation computation (in $\mathbb{G}$ or $\mathbb{G}_T$), respectively.

same concern. An overview comparing our work to their efforts [4, 23] is given in Table 1.

The first CP-ABE construction proposed by Bethencourt et al. [7] was proven secure under the generic group model. Later, Cheung and Newport [16] presented a CP-ABE scheme that is secure under the standard model; however, the access policies in that scheme are restricted to be in the form of a AND combination of different attributes. Recently, secure and more expressive CP-ABE schemes [36, 25, 29, 18, 19] were proposed. Rouselakis and Waters [33] proposed the first unbounded CP-ABE scheme where the public parameters do not impose additional limitations on the functionality of the scheme. Müller et al. [31] and Lewko et al. [27] led another line of research, considering CP-ABE schemes with multiple authorities, in an attempt to meet the need of a more general framework where data are shared according to policies defined over attributes or credentials issued across different trust domains and organizations. Similar to KP-ABE, in virtually all existing CP-ABE schemes, the size of a ciphertext is proportional to the size of its associated access policy, and the decryption cost is proportional to the number of attributes that have been used for decryption. Emura et al. [17] suggested a CP-ABE scheme with constant-size ciphertexts but policies are restricted to a single AND gate. Herranz et al. [22] described a CP-ABE scheme with constant-size ciphertexts, only supporting threshold access policies. Recently, Hohenberger and Waters [23] observed that if one is willing to consider "bounded" CP-ABE systems, where a value k_{max} can be set system-wide as the maximum number of times a single attribute can appear in a particular formula (or access structure), then one can achieve fast decryption without an increase in ciphertext size or encryption time.

Besides the two usual flavors of ABE, another kind of ABE schemes [1], called dual-policy ABE, mixes features from both KP-ABE and CP-ABE schemes.

[2]The measurement is in terms of $|\mathbb{G}|$.

[3]The scheme proposed by Attrapadung et al. [4] supports non-monotonic access structures. We take their KP-ABE construction supporting monotonic access structures for comparison.

[4]In [23], Hohenberger and Waters mentioned that one may modify their construction to achieve full security using dual encryption system [26].

1.3 Organization

The rest of this paper is organized as follows. Section 2 gives some preliminaries and formal definition of KP-ABE. Section 3 describes the proposed construction and its security proof. Section 4 concludes the paper.

2. PRELIMINARIES

If S is a set, then $|S|$ denotes its size and $s_1, \ldots, s_t \leftarrow S$ denotes the operation of picking elements $s_1, \ldots, s_t$ uniformly at random from S. Let $\mathbb{N}$ denote the set of natural numbers. If $n \in \mathbb{N}$ then $[n]$ denotes the set $\{1, \ldots, n\}$. 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 [5]). *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, monotonic access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}} \setminus \{\emptyset\}$. 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 monotonic 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 [5]:

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 [5] 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 $\mathbf{A}$ [5]. 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 [5] 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 [27] for a discussion on how to perform this conversion.

2.3 Key-Policy Attribute-Based Encryption

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

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

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathbb{A})$ takes as input the public parameters PK, the master secret key MSK and an access structure $\mathbb{A}$. It outputs a private key $\mathsf{SK}_\mathbb{A}$ corresponding to $\mathbb{A}$.

$\mathsf{Encrypt}(\mathsf{PK}, M, S)$ takes as input the public parameters PK, a message M and a set of attributes S. It outputs a ciphertext CT.

$\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_\mathbb{A}, CT)$ takes as input the public parameters PK, a private key $\mathsf{SK}_\mathbb{A}$, and a ciphertext CT associated with a set of attributes S. If the set S of attributes satisfies the access structure $\mathbb{A}$, then the algorithm will decrypt the ciphertext and return a message M.

We now give the full security definition for KP-ABE schemes. This is described by a seucirty game between a challenger and an adversary. The game proceeds as follows:

Setup The challenger runs Setup to obtain the public parameters PK and a master secret key MSK. It gives the public parameters PK to the adversary and keeps MSK to itself.

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

Challenge The adversary submits two (equal length) messages M_0, M_1 and a set of attributes S, subject to the restriction that S cannot satisfy any of the queried access structures in Query phase 1. The challenger selects a random bit $\beta \in \{0, 1\}$, sets $CT = \mathsf{Encrypt}(\mathsf{PK}, M_\beta, S)$ and sends CT to the adversary as the challenge ciphertext.

Query phase 2 The adversary continues to adaptively query the challenger for secret keys corresponding to access structures with the restriction that none of these can be satisfied by S.

Guess The adversary outputs its guess $\beta' \in \{0, 1\}$ for β.

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 3. *A key-policy attribute-based encryption scheme is (fully) secure if all PPT adversaries have at most a negligible advantage in the above security game.*

2.4 Composite Order Bilinear Groups

We will construct our scheme in composite order bilinear groups whose order is the product of three distinct primes. Composite order bilinear groups were first introduced in [11].

Let $\mathcal{G}$ be an algorithm that takes as input a security parameter 1^λ and outputs a tuple $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e)$, where p_1, p_2, p_3 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1 p_2 p_3$, 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 $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ to denote the subgroups of $\mathbb{G}$ having order p_1, p_2, p_3, respectively. Observe that $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$. Note also that if $g_1 \in \mathbb{G}_{p_1}$ and $g_2 \in \mathbb{G}_{p_2}$ then $e(g_1, g_2) = 1$. The same rule holds whenever e is applied to elements in distinct subgroups.

We now state the complexity assumptions we use. Assumptions 1, 2 and 3 have already been used in [25, 26]. Utilizing the theorems proposed in [24], one can easily prove that the assumptions hold in the generic group model.

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

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

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

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_1 X_2, Y_2 Y_3, X_3),$$

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

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

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

DEFINITION 4. *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:*

$$(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3,$$

$$g \leftarrow \mathbb{G}_{p_1},\ X_3 \leftarrow \mathbb{G}_{p_3},$$

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

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

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 5. *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:*

$$(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3,$$

$$\alpha, s \leftarrow \mathbb{Z}_N,\ g \leftarrow \mathbb{G}_{p_1}, g_2, X_2, Y_2 \leftarrow \mathbb{G}_{p_2},$$

$$X_3 \xleftarrow{\$} \mathbb{G}_{p_3},$$

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

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

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

3. OUR PROPOSED CONSTRUCTION

In this section, drawing on the hierarchical identity-based encryption (HIBE) scheme with constant-size ciphertexts by Boneh et al. [9] and the KP-ABE scheme by Lewko et al. [25], we present a KP-ABE scheme with constant-size ciphertexts and fast decryption.

Our construction supports arbitrary monotonic access formulas. As in [25], we express access formulas by an LSSS 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 in the system, such as "from", "to", "subject" and "date" in the email example. For simplicity, we also make the following assumptions:

- We assume that the same value never appears in two distinct attribute categories. This requirement can be satisfied easily. For example, by prepending the values with the name of the attribute category they belonging to, the value H("from: Bob") belongs to the attribute category "from" and can not be confused with the value H("to: Bob") that belongs to the attribute category "to", where H is a collision-resistant hash function.
- We assume that each set of attributes associated with a ciphertext has n attributes exactly with each attribute belonging to a different category. In the email example, we may assign the value H("subject: NULL") in the attribute category "subject" to emails that have no subject.

For notational purposes, let i denote the attribute name of the i^{th} attribute category. Each set of attributes S associated with a ciphertext can be parsed as $(z_1, \ldots, z_n)$, where $z_i \in \mathbb{Z}_N$ is the value of attribute name i. We express an access formula by $(\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is $\ell \times m$ 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 name $\rho(i)$ specified by the access formula.

Using our notations, a set of attributes $S = (z_1, \ldots, z_n)$ 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

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

where A_i denotes the i^{th} row of $\mathbf{A}$.

Similar to the KP-ABE scheme in [25], our proposed KP-ABE scheme has the restriction that each attribute name can only be used once in an access structure, which is called one-use KP-ABE. We can obtain a secure KP-ABE scheme where attribute names being used multiple times (up to a constant number of uses fixed at setup) from the one-use scheme by applying the generic transformation given in Lewko et al. [25]. The transformation does not affect the features of our proposed KP-ABE scheme, i.e., constant-size ciphertexts and fast decryption. On the other hand, utilizing the new proof methods proposed by Lewko and Waters [29] recently, it may allow our proposed KP-ABE scheme unrestricted use of attribute names.

Concretely, the proposed KP-ABE scheme consists of the following algorithms:

$\mathsf{Setup}(1^\lambda)$ The setup algorithm first runs $\mathcal{G}(1^\lambda)$ to obtain $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e)$ with $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3$. Next it chooses $g, h_0, h_1, \ldots, h_n \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$ and $\alpha \in \mathbb{Z}_N$ uniformly at random. The public parameters are published as $\mathsf{PK} = (\mathbb{G}, \mathbb{G}_T, e, N, g, h_0, h_1, \ldots, h_n, e(g, g)^\alpha)$. The master secret key is $\mathsf{MSK} = (\alpha, X_3)$.

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathbb{A} = (\mathbf{A}, \rho, \mathcal{T}))$ $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$. The key generation algorithm chooses a random vector $v \in \mathbb{Z}_N^m$ such that $\mathbf{1} \cdot v = \alpha$. (Here, $\mathbf{1}$ denotes the vector with the first entry equal to 1 and the rest equal to 0). Let Q_i denote the set $[n] \backslash \{\rho(i)\}$ for each $i \in [\ell]$. For each row A_i of $\mathbf{A}$, it chooses a random $r_i \in \mathbb{Z}_N$ and random elements $R_i, R'_i, \{R_{i,j}\}_{j \in Q_i} \in \mathbb{G}_{p_3}$ (this is done by raising X_3 to a random power). The secret key $\mathsf{SK}_{\mathbb{A}} = ((\mathbf{A}, \rho, \mathcal{T}),\ \{D_i, D'_i, \{D_{i,j}\}_{j \in Q_i}\}_{i \in [\ell]})$ is computed as

$$D_i = g^{A_i \cdot v} (h_0 h_{\rho(i)}^{t_{\rho(i)}})^{r_i} R_i,\ D'_i = g^{r_i} R'_i,\ D_{i,j} = h_j^{r_i} R_{i,j}.$$

$\mathsf{Encrypt}(\mathsf{PK}, M \in \mathbb{G}_T, S = (z_1, \ldots, z_n) \in \mathbb{Z}_N^n)$ The encryption algorithm chooses $s \in \mathbb{Z}_N$ uniformly at random. The

ciphertext $CT = (S, C, C_0, C_1)$ is computed as

$$C = M \cdot e(g,g)^{\alpha s},\ C_0 = g^s,\ C_1 = \left(h_0 \prod_{i=1}^{n} h_i^{z_i}\right)^s.$$

$\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_{\mathbb{A}}, CT)$ Let $CT = (S = (z_1, \ldots, z_n), C, C_0, C_1)$ and $\mathsf{SK}_{\mathbb{A}} = ((\mathbf{A}, \rho, \mathcal{T}),\ \{D_i,\ D'_i, \{D_{i,j}\}_{j \in Q_i}\}_{i \in [\ell]})$, where $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^{\ell}$. If S satisfies $\mathbb{A}$, the decryption algorithm first finds $\mathcal{I} \subseteq [\ell]$ and constants $\{\omega_i\}_{i \in \mathcal{I}}$ such that

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

Let Q_i denote the set $[n] \backslash \{\rho(i)\}$ for each $i \in \mathcal{I}$. Next, the decryption algorithm computes

$$\widetilde{D}_i = D_i \prod_{j \in Q_i} D_{i,j}^{z_j},\ \text{for each } i \in \mathcal{I}.$$

Note that, if $j \in Q_i$, then $j \neq \rho(i)$. Since for each $i \in \mathcal{I}$, $t_{\rho(i)} = z_{\rho(i)}$, then we have

$$\widetilde{D}_i = g^{A_i \cdot v} \left(h_0 \prod_{j=1}^{n} h_j^{z_j}\right)^{r_i} \cdot \widetilde{R}_i.$$

where $\widetilde{R}_i = R_i \prod_{j=1}^{n} R_{i,j}^{z_j}$.

It then computes:

$$e(C_0, \prod_{i \in \mathcal{I}} \widetilde{D}_i^{\omega_i}) / e(C_1, \prod_{i \in \mathcal{I}} D_i'^{\omega_i}) = e(g,g)^{\alpha s}.$$

The message can then be recovered as $C / e(g,g)^{\alpha s}$.

3.1 Security

We now state the security theorem of our KP-ABE scheme.

THEOREM 1. *If Assumptions 1, 2 and 3 hold, then the proposed KP-ABE scheme is secure.*

Proof. Following the approach by Lewko and Waters [26], 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 $CT = (S, C, C_0, C_1)$. Then, we choose a random exponent $c \in \mathbb{Z}_N$. We also choose a random value $\eta \in \mathbb{Z}_N$. The semi-functional ciphertext CT' is set to be

$$(S, C,\ C_0 \cdot g_2^c,\ C_1 \cdot g_2^{c\eta}).$$

Semi-functional Key A semi-functional key for an access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^{\ell}$, will take on one of two forms. To create a semi-functional key, we first use the key generation algorithm to form a normal secret key $\mathsf{SK}_{\mathbb{A}} = ((\mathbf{A}, \rho, \mathcal{T}),\ \{D_i, D'_i, \{D_{i,j}\}_{j \in Q_i}\}_{i \in [\ell]})$, where Q_i denote the set $[n] \backslash \{\rho(i)\}$. Then, we choose $\eta_0, \eta_1, \ldots, \eta_n$ uniformly at random, and random values $\gamma_i \in \mathbb{Z}_N$ associated with row i of $\mathbf{A}$. We also choose a random vector $w \in \mathbb{Z}_N^m$. The semi-functional key of type 1 is set as

$$((\mathbf{A}, \rho, \mathcal{T}),\ \{D_i \cdot g_2^{A_i \cdot w + \gamma_i(\eta_0 + \eta_{\rho(i)} t_{\rho(i)})},\ D'_i \cdot g_2^{\gamma_i},\ \{D_{i,j} \cdot g_2^{\gamma_i \eta_i}\}_{j \in Q_i}\}_{i \in [\ell]}).$$

A semi-functional key of type 2 is formed without the terms $g_2^{\gamma_i(\eta_0 + \eta_{\rho(i)} t_{\rho(i)})}, g_2^{\gamma_i}, g_2^{\gamma_i \eta_i}$ (one could also interpret this as setting $\gamma_i = 0$):

$$((\mathbf{A}, \rho, \mathcal{T}),\ \{D_i \cdot g_2^{A_i \cdot w},\ D'_i,\ \{D_{i,j}\}_{j \in Q_i}\}_{i \in [\ell]}).$$

Let q denote the number of secret key queries made by the adversary. We will prove the security of our scheme based on Assumptions 1, 2 and 3 using a hybrid argument over a sequence of games.

$\mathsf{Game}_{\mathsf{real}}$ The real security game.

$\mathsf{Game}_{\mathsf{restricted}}$ Let $CT = (S, C, C_0, C_1)$ be the challenge ciphertext, where $S = (z_1, \ldots, z_n)$. This game is the same as $\mathsf{Game}_{\mathsf{real}}$ except that the challenger outputs reject and halts if event E_1 happens. Event E_1 is defined as: the adversary issues a key query $\langle \mathbb{A} = (\mathbf{A}, \rho, \mathcal{T}) \rangle$, where $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^{\ell}$, such that $\exists i \in [n], j \in [\ell]$,

$$z_i \not\equiv t_{\rho(j)} \mod N,\ \text{ and }\ z_i \equiv t_{\rho(j)} \mod p_2.$$

$\mathsf{Game}_{\mathsf{ch}}$ This game is the same as $\mathsf{Game}_{\mathsf{restricted}}$ except that the challenge ciphertext is replaced with a semi-functional ciphertext.

$\mathsf{Game}_{k,1} (1 \leq k \leq q)$ This game is like $\mathsf{Game}_{\mathsf{ch}}$ except for the way that the challenger answers the adversary's secret key queries. The first $k-1$ secret keys are semi-functional of type 2, the k^{th} secret key is semi-functional of type 1, and the remaining secret keys are normal.

$\mathsf{Game}_{k,2} (0 \leq k \leq q)$ This game is like $\mathsf{Game}_{\mathsf{ch}}$ except for the way that the challenger answers the adversary's secret key queries. The first k secret keys are semi-functional of type 2, and the remaining secret keys are normal.

$\mathsf{Game}_{\mathsf{Final}}$ This game is like $\mathsf{Game}_{q,2}$ except that the challenge ciphertext is a semi-functional ciphertext of a random message chosen from $\mathbb{G}_T$.

We prove that these games are computationally indistinguishable in the following four lemmas. Note that $\mathsf{Game}_{\mathsf{ch}} = \mathsf{Game}_{0,2}$. In $\mathsf{Game}_{\mathsf{Final}}$, it is clear that the value of β is information-theoretically hidden from the adversary. Hence the adversary has no advantage in $\mathsf{Game}_{\mathsf{Final}}$. Therefore, we conclude that the advantage of the adversary in $\mathsf{Game}_{\mathsf{real}}$ (i.e., the real security game) is negligible. This completes the proof of Theorem 1.

LEMMA 1. *Suppose that $\mathcal{G}$ satisfies Assumption 1. Then $\mathsf{Game}_{\mathsf{real}}$ and $\mathsf{Game}_{\mathsf{restricted}}$ are computationally indistinguishable.*

Proof. Observe that, if event E_1, which is defined in the description of $\mathsf{Game}_{\mathsf{Restricted}}$, does not happen, $\mathsf{Game}_{\mathsf{Restricted}}$ is identical to $\mathsf{Game}_{\mathsf{Real}}$. All we have to do is to prove that E_1 happens with negligible probability.

If E_1 happens with non-negligible probability, we construct a PPT algorithm $\mathcal{B}$ that breaks Assumption 1 with non-negligible probability. Observe that, given $\mathbb{G}, \mathbb{G}_T, e, N, g, X_1X_2, Y_2Y_3, X_3, T$, algorithm $\mathcal{B}$ can perfectly simulate $\mathsf{Game}_{\mathsf{Real}}$.

Let $CT = (S, C, C_0, C_1)$ be the challenge ciphertext, where $S = (z_1, \ldots, z_n)$. During the simulation, for each key query $\langle \mathbb{A} = (\mathbf{A}, \rho, \mathcal{T}) \rangle$ issued by the adversary, where $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$, if $z_i \not\equiv t_{\rho(j)} \mod N$ for $i \in [n]$ and $j \in [\ell]$, $\mathcal{B}$ computes $a = \gcd(z_i - t_{\rho(j)}, N)$. $\mathcal{B}$ identifies the occurrence of E_1 (i.e., p_2 divides a) with $e(X_1X_2, Y_2Y_3)^a = 1_{\mathbb{G}_T}$. Set $b = \frac{N}{a}$. We consider two cases: 1. p_1 divides b; 2. p_3 divides b.

$\mathcal{B}$ can determine if case 1 has occurred by testing if $e(X_1X_2, g)^b = 1_{\mathbb{G}_T}$. If this happens, $\mathcal{B}$ can then learn whether T has a $\mathbb{G}_{p_2}$ component or not by testing if $e(T,\ X_1X_2)^b = 1_{\mathbb{G}_T}$. If not, then T has a $\mathbb{G}_{p_2}$ component, i.e., $T \in \mathbb{G}_{p_1p_2p_3}$; otherwise, $T \in \mathbb{G}_{p_1p_3}$.

$\mathcal{B}$ can determine if case 2 has occurred by testing if $e(Y_2Y_3, X_3)^b = 1_{\mathbb{G}_T}$. If this happens, $\mathcal{B}$ can then learn whether T has a $\mathbb{G}_{p_2}$ component or not by testing if $e(T, Y_2Y_3)^b = 1_{\mathbb{G}_T}$. If not, then T has a $\mathbb{G}_{p_2}$ component, i.e., $T \in \mathbb{G}_{p_1p_2p_3}$; otherwise, $T \in \mathbb{G}_{p_1p_3}$.

LEMMA 2. *Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathsf{Game}_{\mathsf{restricted}}$ and $\mathsf{Game}_{\mathsf{ch}}$ are computationally indistinguishable.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{restricted}}$ and $\mathsf{Game}_{\mathsf{ch}}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 2. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N, g, X_3, T$ and will simulate $\mathsf{Game}_{\mathsf{restricted}}$ or $\mathsf{Game}_{\mathsf{ch}}$ with $\mathcal{A}$. $\mathcal{B}$ first chooses $\alpha, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ uniformly at random. It then sets $h_0 = g^{a_0}, h_1 = g^{a_1}, \ldots, h_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (\mathbb{G}, \mathbb{G}_T, e, N, g, h_0, h_1, \ldots, h_n, e(g,g)^\alpha).$$

$\mathcal{B}$ can generate normal secret keys in response to $\mathcal{A}$'s key requests by using the key generation algorithm KeyGen, since it knows the master secret key $\mathsf{MSK} = (\alpha, X_3)$ associated with PK. (Notice that, if the event E_1 which is described in $\mathsf{Game}_{\mathsf{restricted}}$ happens when $\mathcal{A}$ makes a key query, $\mathcal{B}$ responds as in $\mathsf{Game}_{\mathsf{restricted}}$.)

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and a set of attributes S. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. Parse S as $(z_1, \ldots, z_n)$. $\mathcal{B}$ computes
$$C = M_\beta \cdot e(g^\alpha, T),\ C_0 = T,\ C_1 = T^{a_0 + \sum_{i=1}^n a_i z_i}.$$
2. $\mathcal{B}$ sets the challenge ciphertext as $CT = (S, C, C_0, C_1)$ and sends it to $\mathcal{A}$.

If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, let $T = g^s g_2^c$, then
$$C = M_\beta \cdot e(g,g)^{\alpha s},$$
$$C_0 = g^s \cdot g_2^c,\ C_i = \left(h_0 \prod_{i=1}^n h_i^{z_i}\right)^s \cdot g_2^{c\eta}.$$
where $\eta = a_0 + \sum_{i=1}^n a_i z_i$. This is a semi-functional ciphertext and $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{ch}}$. We note that the values of a_0, a_i, z_i modulo p_1 are uncorrelated from their values modulo p_2, so this is properly distributed. If $T \leftarrow \mathbb{G}_{p_1}$, it is easy to observe that this is a normal ciphertext and $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{restricted}}$. 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 1. Then for each $k \in [q]$, $\mathsf{Game}_{k-1,2}$ and $\mathsf{Game}_{k,1}$ are computationally indistinguishable.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes $\mathsf{Game}_{k-1,2}$ and $\mathsf{Game}_{k,1}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 1. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N, g, X_1X_2, Y_2Y_3, X_3, T$ and will simulate $\mathsf{Game}_{k-1,2}$ or $\mathsf{Game}_{k,1}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $\alpha, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ uniformly at random. It then sets $h_0 = g^{a_0}, h_1 = g^{a_1}, \ldots, h_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (\mathbb{G}, \mathbb{G}_T, e, N, g, h_0, h_1, \ldots, h_n, e(g,g)^\alpha).$$

Note that $\mathcal{B}$ knows the master secret key $\mathsf{MSK} = (\alpha, X_3)$ associated with PK. Let us now explain how $\mathcal{B}$ answers the μ-th key query for an access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$. Let Q_i denote the set $[n] \backslash \{\rho(i)\}$ for each $i \in [\ell]$. (Notice that, if the event E_1 described in $\mathsf{Game}_{\mathsf{restricted}}$ happens when $\mathcal{A}$ makes a key query, $\mathcal{B}$ responds as in $\mathsf{Game}_{\mathsf{restricted}}$.)

For $\mu < k$, $\mathcal{B}$ creates a semi-functional key of type 2 by choosing a random vector v such that $\mathbf{1} \cdot v = \alpha$, a random vector $w' \in \mathbb{Z}_N^m$, random exponents $r_i \in \mathbb{Z}_N$, random elements $R_i, R_i', \{R_{i,j}\}_{j \in Q_i} \in \mathbb{G}_{p_3}$, and for each $i \in [\ell], j \in Q_i$ setting:

$$D_i = g^{A_i \cdot v} (h_0 h_{\rho(i)}^{t_{\rho(i)}})^{r_i} R_i (Y_2Y_3)^{A_i \cdot w'},\ D_i' = g^{r_i} R_i',$$
$$D_{i,j} = h_j^{r_i} R_{i,j},$$

We note that this is a properly distributed semi-functional key of type 2 because the value of $A_i \cdot w'$ modulo p_2 is uncorrelated to its value modulo p_3.

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

To answer the k-th key quest for $(\mathbf{A}, \rho, \mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}))$, $\mathcal{B}$ chooses a random vector $v' \in \mathbb{Z}_N^m$ such that $v' \cdot \mathbf{1} = \alpha$, a random vector $w \in \mathbb{Z}_N^m$ such that $w \cdot \mathbf{1} = 0$, random exponents $\tilde{\gamma}_i \in \mathbb{Z}_N$, random elements $\tilde{R}_i, \tilde{R}_i', \{\tilde{R}_{i,j}\}_{j \in Q_i} \in \mathbb{G}_{p_3}$ and for each $i \in [\ell], j \in Q_i$ sets:

$$D_i = g^{A_i \cdot v'} T^{A_i \cdot w} T^{\tilde{\gamma}_i (a_0 + a_{\rho(i)} t_{\rho(i)})} \tilde{R}_i,$$
$$D_i' = T^{\tilde{\gamma}_i} \tilde{R}_i',\ D_{i,j} = T^{\tilde{\gamma}_i a_j} \tilde{R}_{i,j}.$$

We have the following observations. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then T can be written as $g^r g_2^d R$, and

$$D_i = g^{A_i \cdot v} (h_0 h_{\rho(i)}^{t_{\rho(i)}})^{r_i} R_i \cdot g_2^{\delta_i + \gamma_i (\eta_0 + \eta_{\rho(i)} t_{\rho(i)})},$$
$$D_i' = g^{r_i} R_i' \cdot g_2^{\gamma_i},\ D_{i,j} = h_j^{r_i} R_{i,j} \cdot g_2^{\gamma_i \eta_j}$$

where $v = v' + rw$, $r_i = r\tilde{\gamma}_i$, $\delta_i = dA_iw$, $\gamma_i = d\tilde{\gamma}_i$, $R_i = R^{A_i w + \tilde{\gamma}_i (a_0 + a_{\rho(i)} t_{\rho(i)})} \tilde{R}_i$, $R_i' = R^{\tilde{\gamma}_i} \tilde{R}_i'$, $R_{i,j} = R^{\tilde{\gamma}_i a_j} \tilde{R}_{i,j}$, $\{\eta_l = a_l\}_{l \in [n]}$. This is a semi-function key of type 1. Note that the values of $\tilde{\gamma}_i, a_0, a_i, t_{\rho(i)}$ modulo p_1 are uncorrelated from their values modulo p_2. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed normal secret key.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and a set of attributes S. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. Parse S as $(z_1, \ldots, z_n)$. $\mathcal{B}$ computes

$$C = M_\beta \cdot e(g^\alpha, X_1X_2),\ C_0 = X_1X_2,$$
$$C_1 = (X_1X_2)^{a_0+\sum_{i=1}^n a_i z_i},$$

2. $\mathcal{B}$ sets the challenge ciphertext as $CT = (S, C, C_0, C_1)$ and sends it to $\mathcal{A}$.

If we write X_1X_2 as $g^s g_2^c$, then

$$C = M_\beta \cdot e(g,g)^{\alpha s},\ C_0 = g^s \cdot g_2^c,$$
$$C_1 = \left(h_0 \prod_{i=1}^n h_i^{z_i}\right)^s \cdot g_2^{c\eta}.$$

where $\eta = a_0 + \sum_{i=1}^n a_i z_i$. This is a semi-functional ciphertext. Note that the values of a_0, a_i, z_i modulo p_1 are uncorrelated from their values modulo p_2.

Next, we show that the k^{th} key and the challenge ciphertext are properly distributed. Observe that, if $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then the k^{th} key is a properly distributed normal secret key and the challenge ciphertext is a properly distributed semi-functional ciphertext. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, since the event E_1 happens with negligible probability which has been proven in Lemma 1, then with overwhelming probability, there exits $i \in [\ell]$ such that $t_{\rho(i)} \not\equiv z_{\rho(i)} \mod N$ and $t_{\rho(i)} \mod p_2$ is uniformly distributed[5]; thus similar to the analysis in the proof of Lewko *et al.* 's KP-ABE scheme [25], the k^{th} key and the challenge ciphertext are properly distributed.

We can thus conclude that, if $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,1}$. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k-1,2}$. 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 1. Then for each $k \in [q]$, $\mathsf{Game}_{k,1}$ and $\mathsf{Game}_{k,2}$ are computationally indistinguishable.*

Proof. Suppose there exists an adversary $\mathcal{A}$ 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 1. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N, g, X_1X_2, Y_2Y_3, X_3, T$ and will simulate $\mathsf{Game}_{k,1}$ or $\mathsf{Game}_{k,2}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $\alpha, a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ uniformly at random. It then sets $h_0 = g^{a_0}, h_1 = g^{a_1}, \ldots, h_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (\mathbb{G}, \mathbb{G}_T, e, N, g, h_0, h_1, \ldots, h_n, e(g,g)^\alpha).$$

The responses to all key queries and challenge ciphertexts are the same as in Lemma 3, except that the k-th key query which is given below.

To answer the k-th key quest for $(\mathbf{A}, \rho, \mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}))$, $\mathcal{B}$ chooses a random vector $v \in \mathbb{Z}_N^m$ such that $v \cdot \mathbf{1} = \alpha$, a random vector $w \in \mathbb{Z}_N^m$, random exponents $\tilde{\gamma}_i \in \mathbb{Z}_N$, random elements $\tilde{R}_i, \tilde{R}_i', \{\tilde{R}_{i,j}\}_{j \in Q_i} \in \mathbb{G}_{p_3}$ and for each $i \in [\ell], j \in Q_i$ sets:

$$D_i = g^{A_i \cdot v} (Y_2Y_3)^{A_i \cdot w} T^{\tilde{\gamma}_i (a_0 + a_{\rho(i)} t_{\rho(i)})} \tilde{R}_i,$$
$$D_i' = T^{\tilde{\gamma}_i} \tilde{R}_i',\ D_{i,j} = T^{\tilde{\gamma}_i a_j} \tilde{R}_{i,j}.$$

[5]Notice that, We have already assumed that the same value never appears in two distinct attribute categories and each attribute name/category can only be used once in an access structure. That is, for each $i_1, i_2 \in [n]$ and $j_1, j_2 \in [\ell]$, if $i_1 \neq i_2, j_1 \neq j_2$, then $z_{i_1} \neq z_{i_2}$ and $t_{\rho(j_1)} \neq t_{\rho(j_2)}$.

We have the following observations. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then T can be written as $g^r g_2^d R$, and

$$D_i = g^{A_i \cdot v} (h_0 h_{\rho(i)}^{t_{\rho(i)}})^{r_i} R_i \cdot g_2^{\delta_i + \gamma_i (\eta_0 + \eta_{\rho(i)} t_{\rho(i)})},$$
$$D_i' = g^{r_i} R_i' \cdot g_2^{\gamma_i},\ D_{i,j} = h_j^{r_i} R_{i,j} \cdot g_2^{\gamma_i \eta_j},$$

where $r_i = r\tilde{\gamma}_i$, $\delta_i = \log_{g_2} Y_2 \cdot A_i w$, $\gamma_i = d\tilde{\gamma}_i$, $R_i = Y_3^{A_i \cdot w} \cdot R^{\tilde{\gamma}_i (a_0 + a_{\rho(i)} t_{\rho(i)})} \tilde{R}_i$, $R_i' = R^{\tilde{\gamma}_i} \tilde{R}_i'$, $R_{i,j} = R^{\tilde{\gamma}_i a_j} \tilde{R}_{i,j}$, $\{\eta_l = a_l\}_{l \in [n]}$. This is a semi-function key of type 1. Note that the values of $\tilde{\gamma}_i, a_0, a_i, t_{\rho(i)}$ modulo p_1 are uncorrelated from their values modulo p_2. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 2.

Similar to the above lemma, We can conclude that, if $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,1}$. If $T \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, 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 $\mathsf{Game}_{q,2}$ and $\mathsf{Game}_{\mathsf{Final}}$ are computationally indistinguishable.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes $\mathsf{Game}_{q,2}$ and $\mathsf{Game}_{\mathsf{Final}}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 3. $\mathcal{B}$ is given $(\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g^\alpha X_2, g^s Y_2, X_3, T)$ and will simulate $\mathsf{Game}_{q,2}$ or $\mathsf{Game}_{\mathsf{Final}}$ with $\mathcal{A}$. $\mathcal{B}$ chooses $a_0, a_1, \ldots, a_n \in \mathbb{Z}_N$ uniformly at random. It then sets $h_0 = g^{a_0}, h_1 = g^{a_1}, \ldots, h_n = g^{a_n}$, and sends $\mathcal{A}$ the public parameters:

$$\mathsf{PK} = (\mathbb{G}, \mathbb{G}_T, e, N, g, h_0, h_1, \ldots, h_n, e(g^\alpha X_2, g) = e(g,g)^\alpha).$$

Each time $\mathcal{B}$ is asked to provide a secret key for an access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is an $\ell \times m$ matrix, ρ is a map from each row A_i of $\mathbf{A}$ to $\{1, \ldots, n\}$ and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$, $\mathcal{B}$ creates a semi-functional key of type 2 as follows.

1. $\mathcal{B}$ chooses $v_2, \ldots, v_m \in \mathbb{Z}_N$ randomly and implicity sets $v = (\alpha, v_2, \ldots, v_m) \in \mathbb{Z}_N^m$. Note that $\mathbf{1} \cdot v = \alpha$. $\mathcal{B}$ also chooses a random vector $w' \in \mathbb{Z}_N^m$

2. For each $i \in [\ell]$, $\mathcal{B}$ random exponents $r_i \in \mathbb{Z}_N$ and random elements $R_i, R_i', \{R_{i,j}\}_{j \in Q_i} \in \mathbb{G}_{p_3}$, where Q_i denote the set $[n] \backslash \{\rho(i)\}$.

3. For each $i \in [\ell]$, let $A_i = (A_{i,1}, \ldots, A_{i,m}) \in \mathbb{Z}_N^m$, where A_i is the row i of $\mathbf{A}$. $\mathcal{B}$ sets the secret key $\mathsf{SK}_\mathbb{A}$ as $((\mathbf{A}, \rho, \mathcal{T}),\ \{D_i, D_i', \{D_{i,j}\}_{j \in Q_i}\}_{i \in [\ell]})$ and sends it to $\mathcal{A}$, where

$$D_i = (g^\alpha X_2)^{A_{i,1}} g^{\sum_{l=2}^m A_{i,l} v_l} g_2^{A_i \cdot w'} \cdot (h_0 h_{\rho(i)}^{t_{\rho(i)}})^{r_i} R_i,$$
$$D_i' = g^{r_i} R_i',\ D_{i,j} = h_j^{r_i} R_{i,j}.$$

   Observe that, D_i can be written as

$$g^{A_i \cdot v} (h_0 h_{\rho(i)}^{t_{\rho(i)}})^{r_i} R_i \cdot g_2^{A_i \cdot w},$$

   where $w = w'$ except with a $\log_{g_2} X_2$ added in the first coordinate; hence $\mathsf{SK}_\mathbb{A}$ is a properly distributed semi-functional key of type 2.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two (equal length) messages M_0, M_1 and a set of attributes S. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ randomly and does the following.

1. Parse S as $(z_1, \ldots, z_n)$. $\mathcal{B}$ computes

$$C = M_\beta \cdot T,\ C_0 = g^s Y_2,\ C_1 = (g^s Y_2)^{a_0 + \sum_{i=1}^n a_i z_i}.$$

2. $\mathcal{B}$ sets the challenge ciphertext as $CT = (S, C, C_0, C_1)$ and sends it to $\mathcal{A}$.

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

$$C_0 = g^s \cdot g_2^c,\ C_1 = \left(h_0 \prod_{i=1}^n h_i^{z_i} \right)^s \cdot g_2^{c\eta},$$

where $\eta = a_0 + \sum_{i=1}^n a_i z_i$. Note that the values of a_0, a_i, z_i modulo p_1 are uncorrelated to their values modulo p_2. Hence, if $T = e(g,g)^{\alpha s}$, then CT is a properly distributed semi-functional ciphertext of M_β. On the other hand, if $T \leftarrow \mathbb{G}_T$, then CT is a properly distributed semi-functional ciphertext of a random message.

We can conclude that, if $T = e(g,g)^{\alpha s}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{q,2}$. If $T \leftarrow \mathbb{G}_T$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Final}}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T.

4. CONCLUSIONS

We presented the first KP-ABE scheme, which has the following features simultaneously: expressive (i.e., supporting arbitrary monotonic access structure); fully secure in the standard model; constant-size ciphertexts and fast decryption. In the future, it will be interesting to see if shorter secret keys can be obtained without affecting the other features of our proposed scheme. Another challenging problem is to design CP-ABE schemes with the similar features of our proposed KP-ABE scheme.

5. ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their helpful comments. The work of Junzuo Lai was supported by the National Natural Science Foundation of China (Nos. 61300226, 61272534, 61272453), the Research Fund for the Doctoral Program of Higher Education of China (No. 20134401120017), the Guangdong Provincial Natural Science Foundation (No. S2013040014826), the Open Research Fund of State Key Lab. of Integrated Services Networks (No. ISN15-04), and the Fundamental Research Funds for the Central Universities. The work of Robert H. Deng was supported by the the research grant 13-C220-SMU-05 from the Office of Research, Singapore Management University. The work of Jian Weng was supported by the National Science Foundation of China (Nos. 61272413, 61373158, 61133014, 61272415), the Fok Ying Tung Education Foundation (No. 131066), the Program for New Century Excellent Talents in University (No. NCET-12-0680), and the Research Fund for the Doctoral Program of Higher Education of China (No. 20134401110011).

6. REFERENCES

[1] N. Attrapadung and H. Imai. Dual-policy attribute based encryption. In *ACNS*, pages 168–185, 2009.

[2] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Public Key Cryptography*, pages 384–402, 2010.

[3] N. Attrapadung and B. Libert. Functional encryption for public-attribute inner products: Achieving constant-size ciphertexts with adaptive security or support for negation. *J. Mathematical Cryptology*, 5(2):115–158, 2012.

[4] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography*, pages 90–108, 2011.

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

[6] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

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

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

[9] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[10] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

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

[12] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.

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

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

[15] 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.

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

[17] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *ISPEC*, pages 13–23, 2009.

[18] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. *IACR Cryptology ePrint Archive*, 2013:128, 2013.

[19] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013.

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

[21] 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.

[22] J. Herranz, F. Laguillaumie, and C. Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In *Public Key Cryptography*, pages 19–34, 2010.

[23] S. Hohenberger and B. Waters. Attribute-based encryption with fast decryption. In *Public Key Cryptography*, pages 162–179, 2013.

[24] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. Cryptology ePrint Archive, Report 2007/404, 2007. `http://eprint.iacr.org/`.

[25] 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.

[26] 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.

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

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

[29] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pages 180–198, 2012.

[30] 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.

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

[32] 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.

[33] Y. Rouselakis and B. Waters. New constructions and proof methods for large universe attribute-based encryption. *IACR Cryptology ePrint Archive*, 2012:583, 2012.

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

[35] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

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