An Efficient KP-ABE with Short Ciphertexts in Prime Order Groups under Standard Assumption

Jongkil Kim
Data61, CSIRO
Australia
jongkil.kim@data61.csiro.au

Willy Susilo
University of Wollongong
Australia
wsusilo@uow.edu.au

Fuchun Guo
University of Wollongong
Australia
fuchun@uow.edu.au

Man Ho Au
The Hong Kong Polytechnic
University, Hong Kong
csallen@comp.polyu.edu.hk

Surya Nepal
Data61, CSIRO
Australia
surya.nepal@csiro.au

ABSTRACT

We introduce an efficient Key-Policy Attribute-Based Encryption (KP-ABE) scheme in prime order groups. Our scheme is semi-adaptively secure under the decisional linear assumption and supports a large universe of attributes and multi-use of attributes. Those properties are critical for real applications of KP-ABE schemes since they enable an efficient and flexible access control. Prior to our work, existing KP-ABE schemes with short ciphertexts were in composite order groups or utilized either Dual Pairing Vector Spaces (DPVS) or Dual System Groups (DSG) in prime order groups. However, those techniques brought an efficiency loss. In this work, we utilize *a nested dual system encryption* which is a variant of Waters' dual system encryption (Crypto' 09) to achieve semi-adaptively secure KP-ABE. As a result, we obtain a new scheme having better efficiency compared to existing schemes while it keeps a semi-adaptive security under the standard assumption. We implement our scheme and compare its efficiency with the previous best work.

Keywords

attribute based encryption; dual system encryption; short ciphertexts; prime order groups; standard assumption

1. INTRODUCTION

Attribute Based Encryption [37] (ABE) is a public key based encryption system which allows users to access secret data based on their attributes. The concept of ABE was refined by Goyal, Pandey, Sahai and Waters [22]. They defined two types of ABE systems, namely, Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE), based on where an access policy is located. In KP-ABE, an access policy is employed in a user's private key while a ciphertext is associated with a set of attributes.

One of the applications of ABE is to provide a fine-grained access control mechanism for large-scale IT systems (e.g. cloud storage) [43, 40, 24]. To use a KP-ABE scheme in a real system such as a cloud storage, efficiency and flexibility must be guaranteed. There exist important properties for KP-ABE to satisfy both of them. In order to satisfy this efficiency requirement, the KP-ABE scheme must be constructed in prime order groups. This means that it has significant benefits not only in terms of low computation costs but also because of the shorter parameters. It is well-known that the pairing computation and exponentiation in prime order groups are more than fifty times faster than those in composite order groups [19]. This performance gap between prime order groups and composite order groups cannot be ignored in a practical usage. Additionally, efficiency is improved further by the achievement of constant-size ciphertexts. The benefits of reducing the size of ciphertexts of the KP-ABE system are due to the fact that they are directly related to the amount of data traffic or the size of the cloud data storage.

To support flexibility, the KP-ABE scheme should be expressive and allow multi-use of attributes. Therefore, it can be used to support any boolean functions for access controls without any pre-processing or encoding to work around the restriction of an appearance of attributes in the access policy to be adapted to any situation while the system is operating and growing. Moreover, the scheme can be more flexible by supporting a large universe of attributes. In KP-ABE with a large universe of attributes, the public parameter does not depend on the total number of attributes in the system. This is useful for a growing IT system. In this setting, the system developer does not need to consider all attributes at the initialization stage. It is also flexible since operators can add new attributes to the system whenever required (simply via allocating unique IDs for new attributes in the system while the system is operating).

1.1 Our Contribution

In this paper, we introduce an expressive semi-adaptively secure KP-ABE scheme in prime order groups under the standard assumptions. Our scheme features both efficiency and flexibility because our scheme has a short ciphertext and supports a large universe of attributes and multi-use

ASIA CCS '17, April 02-06, 2017, Abu Dhabi, United Arab Emirates
© 2017 ACM. ISBN 978-1-4503-4944-4/17/04...$15.00
DOI: http://dx.doi.org/10.1145/3052973.3053003

of attributes. Compared to existing similar schemes, our scheme shows a better efficiency without a loss of flexibility.

Currently, the best scheme is proposed by Agrawal and Chase [3]. Their scheme is based on Attrapadung's scheme [4] which is originally adaptively secure in composite order groups under q-type assumptions. They showed that this scheme was also semi-adaptive secure in prime order groups under standard assumptions. While semi-adaptive security is weaker than the adaptive security, it is still reasonable since the adversary can see the public parameters before it declares the target ciphertext. Their scheme remains the most efficient scheme (in particular, the scheme under the Symmetric External Diffie-Hellman (SXDH) assumption) among the schemes supporting both prime order groups and a large universe.

In this work, we suggest a new KP-ABE scheme with a short ciphertext. Our scheme outperforms Agrawal and Chase's scheme even under a weaker assumption (the Decisional Linear (DLIN) assumption). Simply, for each attribute in a ciphertext and a key ciphertext, Agrawal and Chase's scheme needs two group elements, but ours need only a group element and an integer. In practice, replacing a group element with an integer reduced the size of ciphertexts and keys significantly. For example, in popular Miyaji, Nakabayashi and Takano (MNT) groups [31], the size of an integer is only a half of a group element of G_1 and one-sixth of a group element of G_2 where G_1 and G_2 are left and right inputs of pairing computation e (i.e. $e : G_1 \times G_2 \rightarrow G_T$). Moreover, our scheme has faster encryption and decryption. The number of pairing computations of our scheme is constant while the number of pairing operations of Agrawal and Chase's scheme increases linearly as the number of attributes used in a decryption increases. We implement our scheme and Agrawal and Chase's scheme for comparison. Our scheme reduces the encryption and decryption time by more than 35 percent and 14 percent compared to those of Agrawal and Chase's scheme.

Comparisons with other schemes having short ciphertexts are shown in Table 1. The first expressive KP-ABE with short ciphertexts was introduced by Attrapadung, Libert and Panafieu [7]. However, it is only selectively secure under q-type assumption and supports a small universe of attributes. Later, Attrapadung [4] introduced an adaptively secure KP-ABE scheme with a constant size ciphertext. However, their scheme is inefficient since it was introduced in composite order groups, and it still relies its security on q-type assumptions. More recently, Attrapadung [5] introduced a technique to convert ABE schemes in composite order groups to the schemes in prime order groups. Therefore, the scheme from [4] can be featured into prime order groups while it is still adaptively secure. However, the original construction depends on q-type assumptions. Therefore, the resulting scheme depends also on multiple assumptions including two q-type assumptions. According to Cheon [16] and Sakemi et al. [38], the schemes under q-type assumptions are vulnerable in practice since they can be broken by specific attacks. Takashima [39] uses a sparse DPVS and achieves semi-adaptively security in prime order groups under the standard assumption. Their scheme is more expressive than the other schemes since it supports non-monotone access policy, but their scheme is less efficient than Agrawal and Chase's scheme and ours.

1.2 Our Technique

Water's dual system encryption [41] is a well-known proof technique for public key cryptography. In the dual system encryption, keys and ciphertexts in a construction, namely normal keys and normal ciphertexts, are changed to auxiliary types, namely *semi-functional* keys and *semi-functional* ciphertexts which are only used in a security proof. Semi-functional keys cannot decrypt semi-functional ciphertexts, but they can decrypt normal ciphertexts while normal keys can decrypt both semi-functional ciphertexts and normal ciphertexts. Then, it must be shown that a security game consisting of only semi-functional keys and semi-functional ciphertexts is indistinguishable from a real game which consists of normal keys and normal ciphertext. To do this, first, the challenge ciphertext is changed from normal type to semi-functional type (i.e., *semi-functional ciphertext invariance*). Then, it changes all keys from normal type to semi-functional type one by one (i.e., *semi-functional key invariance*). After the types of all keys and the challenge ciphertext are changed to semi-functional, the proof of the security (*semi-functional security*) is relatively easy since keys cannot decrypt the challenge ciphertext due to their semi-functionality.

In the dual system encryption, proving the semi-functional key invariance is most critical. However, achieving the semi-functional key invariance for ABE schemes is a daunting task. In KP-ABE, a challenge key and the challenge ciphertext have multiple attributes where the challenge key is a key of which the simulator wants to distinguish the type. Additionally, those attributes can appear in both the challenge key and the challenge ciphertext at the same time unless an access structure of the key was not satisfied by a set of attributes for the challenge ciphertext. Moreover, achieving KP-ABE with short ciphertexts in a large attribute universe is more difficult since the total number of attributes in the system is not bounded or exponentially large since we must reuse limited entropy to unbounded attributes.

We solve those problems by utilizing a *nested dual system encryption.* We let a row of a private key denote corresponding key elements of a row of an access matrix. Also, we let a semi-functional row denote a row of which corresponding key elements have the same distribution with those of a semi-functional key. In our security analysis, proving the invariance between a normal key and a semi-functional key is replaced by showing the invariance of a normal row and a semi-functional row. Instead of changing all rows from normal to semi-functional at once, we change each row of an access matrix of the challenge key from normal to semi-functional one-by-one until a normal key turns to a semi-functional key.

The way of converting a normal row to a semi-functional one is similar to that of Waters' Identity Based Encryption (IBE) [41]. In IBE scheme, users have a key based on their identities and the ciphertext is created only for a single identity. We consider IBE scheme as the simplest ABE scheme. In the analysis of our KP-ABE scheme, we can isolate an attribute from other attributes using the nested dual system encryption similar to IBE. This allows us to apply the technique of Waters' IBE to convert the corresponding row to semi-functional.

However, we must solve two problems to apply Waters' technique to our scheme. Firstly, the isolated attribute can appear in the challenge ciphertext. In the security model

Table 1: Comparisons of KP-ABEs with constant size ciphertexts

	ALP11 [7]	A14 [4]	T14 [39]	AC16 [3]	A15 [5]	Ours
Assump.	q-type	q-type, SDs	DLIN	SXDH	q-type, DLIN	DLIN
Universe	small	large	large	large	large	large
Security	Selective	Adaptive	Semi-adap.	Semi-adap.	Adaptive	Semi-adap.
Order	Prime	Composite	Prime	Prime	Prime	Prime
A.S.	NM	M	NM	M	M	M
PK	$O(n)\lvert\mathbb{G}\rvert$	$O(n)\lvert\mathbb{G}\rvert$	$O(n)\lvert\mathbb{G}\rvert$	$(14+2n)\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T\rvert$	$(21+3n)\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T\rvert$	$(12+n)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$
SK	$O(mn)\lvert\mathbb{G}\rvert$	$O(mn)\lvert\mathbb{G}\rvert$	$(5+6mn)\lvert\mathbb{G}\rvert$	$(6+6m+2mn)\lvert\mathbb{G}_2\rvert$	$(9+9m+3mn)\lvert\mathbb{G}_2\rvert$	$(7+n)m\lvert\mathbb{G}\rvert$ $+nm\lvert\mathbb{Z}_p\rvert$
CT	$3\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_T\rvert$	$6\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_T\rvert$	$17\lvert\mathbb{G}\rvert + 1\lvert\mathbb{G}_T\rvert$	$12\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_T\rvert$	$18\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_T\rvert$	$9\lvert\mathbb{G}\rvert + 1\lvert\mathbb{Z}_p\rvert$ $+1\lvert\mathbb{G}_T\rvert$

n: the maximum number of attributes per ciphertext, A.S.*: access structure*
m: the number of rows of an access matrix, *NM : Non-monotone*, *M : Monotone*

of Waters' IBE, the adversary only can query the keys if the identities for the queried keys are different from that of the challenge ciphertext. To overcome this difference, we only change rows of the challenge key from normal to semi-functional if their corresponding attributes are not in the target set of attributes in the challenge ciphertext. Because our scheme is semi-adaptively secure, the simulator knows the target set of attributes before it generates any keys. Therefore, it always can choose those rows in the invariance proof.

Secondly, in our scheme, the challenge ciphertext still has multiple attributes unlike IBE scheme because we only nest attributes in a key, not the challenge ciphertext. Since Waters' IBE uses *pairwise independence* which allows only one attribute in the ciphertext, we need an alternative information theoretical argument for our scheme. We solve this problem by using n-wise independence [6]. If we let A_x denote the x^{th} row of A, Tags for the A_x, $kTag_{j,x}\ \forall j \in [n]$ and a tag for the challenge ciphertext, $cTag$, are generated as

$$\begin{pmatrix} -\rho(x) & 1 & & & \\ -(\rho(x))^2 & & 1 & & \\ \vdots & & & \ddots & \\ -(\rho(x))^n & & & & 1 \\ c_0 & c_1 & c_2 & \cdots & c_n \end{pmatrix} \begin{pmatrix} h'_0 \\ h'_1 \\ h'_2 \\ \vdots \\ h'_n \end{pmatrix} = \begin{pmatrix} kTag_{1,x} \\ kTag_{2,x} \\ \vdots \\ kTag_{n,x} \\ cTag \end{pmatrix}$$

where c_j is coefficients of y^j of $\prod_{\rho(x)\in S^*}(y-\rho(x))$ and S^* is the target set of attributes for the challenge ciphertext. Moreover, we set $h'_0, ...h'_n$ are information theoretically hidden to the adversary. It means that the values of $h'_0, ...h'_n$ are not revealed anywhere else. Therefore, the correlation between tags in the challenge key (in particular, tags of the isolated row in the challenge key) and the challenge ciphertext is information theoretically hidden to the adversary as the dual system encryption requires.

1.3 Related Works

Although the first KP-ABE was introduced by Goyal et al. [22] under the Decisional Bilinear Diffie-Hellman assumption, q-type assumptions were widely used to prove the selective security of Attribute-Based Encryption [42, 36, 7] as in other public key encryption systems [9, 10, 18, 20]. Although Lewko and Waters [29] demonstrated how q-type assumptions were able to be utilized to achieve adaptive security, q-type assumptions are less desirable for the security of cryptographic primitives. Cheon [16] and Sakemi et al. [38] show that schemes under q-type assumptions are vulnerable in practice.

Several fully secure KP-ABE schemes [27, 28, 4] were introduced in composite order groups. However, it is well known that composite order groups bring significant inefficiency into an encryption system. Guillevic [23] noted that to achieve 128 bits security in composite order groups, the size of each group element must be about 10 times larger than the group element of prime order groups. In addition, computing a pairing operation is more than 200 times slower even if the group order is the product of two primes. More recently, in [29, 26], this inefficiency of composite order group was eased through the use of Dual Pairing Vector Spaces (DPVS) [32, 34, 33] technique. Essentially, DVPS allows the construction which exhibits key properties of composite order groups, including parameter hiding and orthogonality, using prime orders by orthogonal vector spaces, but DPVS still retains an efficiency loss caused by the size of a vector since the size of parameters and the number of pairing computations normally increases linearly with the size of the vector in the system. Therefore, it remains a difficult task to construct an ABE achieving more than selective security under static assumptions without using composite order groups or DPVS.

More recently, Dual System Groups (DSG) [13] were often used to construct ABE in prime order groups. In DSG, the size of parameters of ABE scheme depends on the assumptions they relied on. Agrawal and Chase [3] suggested KP-ABE scheme with a short ciphertext using dual system groups using DSG. In their scheme, they showed that Attrapadung's scheme [4] can be featured in prime order groups under the standard assumption. In their conversion, one element in composite order groups [4] can be realized two group elements in prime order groups under the SXDH (Symmetric eXternal Diffie-Hellman) assumption or three group elements under the more standardized Decisional Linear (DLIN) assumption.

The first fully secure expressive KP-ABE scheme was introduced by Lewko, Okamoto, Sahai, Takashima and Waters [27] in composite order groups while a selectively secure scheme was first introduced by Goyal et al. [22] in prime order groups. Since then, KP-ABE of [27] was developed in many directions. KP-ABE schemes with *a large universe* are suggested in [4, 28, 35]. In *a large universe* of attributes, KP-ABE system supports exponentially many attributes. Also, they support a growing system without changing the initial set-up. To take benefits of prime order groups, a scheme

[35] with a large universe also suggested using DPVS. But, it does not allow multi-use of attributes and constant-size ciphertexts.

A semi-adaptive security model is suggested by Chen and Wee [15]. It is a stronger notion than a selective security, but weaker than an adaptive security. In a semi-adaptive secure KP-ABE, the adversary is not required to declare its target before seeing any public parameters, but it must declare the target before it queries any private key.

Kim et al. [25] introduced a generic way (i.e. tag based encoding) to achieve encryption scheme using Water's dual system encryption. They generalized Waters' identity-based encryption [41] for generic encryption schemes but without a nested methodology. Therefore, our scheme cannot be fitted into their framework.

2. BACKGROUND

2.1 Monotone Access Structures [8]

Definition 1 *(Access Structure) Let $\{P_1, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subset 2^{\{P_1,...,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subset C$, then $C \in \mathbb{A}$. An monotone access structure is a monotone collection $\mathbb{A}$ of non-empty subsets of $\{P_1, ..., P_n\}$, i.e., $\mathbb{A} \subset 2^{\{P_1,...,P_n\}} \setminus \{\}$. If the sets in $\mathbb{A}$, they are called the authorized sets. Otherwise, if the sets not in $\mathbb{A}$, they are called the unauthorized sets.*

Definition 2 *(Linear Secret-Sharing Schemes (LSSS)) 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. The share-generating matrix A exists for Π. The matrix A has m rows and ℓ columns. For all $i = 1, ..., m$, the i^{th} row of A is labelled by a party $\rho(x)$ ($\rho : \{1, ..., m\} \to \mathcal{P}$). When we consider the column vector $v = (s, r_2, ..., r_\ell) \in \mathbb{Z}_p^\ell$, where s is the secret to be shared and $r_2, ..., r_\ell$ are randomly selected, then Av is the vector of m shares of the secret s according to Π. The share $(Av)_i$ belongs to party $\rho(x)$.

We point out that, in our KP-ABE scheme, ρ is not necessary to be injective. It means that our KP-ABE scheme naturally allows multi-use of attributes.

2.2 Bilinear Maps in Prime Order Groups

We briefly describe the important properties of prime order bilinear groups. Let set $\mathcal{G}$ as a group generator taking a security parameter λ as input and outputting a description of a bilinear group $\mathcal{G}$. For our purposes, we will have $\mathcal{G}$ output (p, G_1, G_2, G_T, e) where p are a prime, G_1, G_2 and G_T are cyclic groups of order p, and $e : G_1 \times G_2 \to G_T$ is an efficiently computable non-degenerate bilinear map.

We assume that the group operations in G_1, G_2 and G_T as well as the bilinear map e are efficiently computable in polynomial time with respect to λ and that the group descriptions of G_1, G_2 and G_T include generators of the respective cyclic groups. If $G_1 = G_2$, we call e a symmetric pairing and we use G to denote both G_1 and G_2 (i.e. $e : G \times G \to G_T$). Otherwise, we call e an asymmetric pairing.

2.3 Complexity Assumptions

Decisional Linear Assumption (*DLIN*) Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (p, G, G_T, e) \xleftarrow{R} \mathcal{G}, \quad g, f, \nu \xleftarrow{R} G, \quad c_1, c_2 \xleftarrow{R} \mathbb{Z}_p,$$

$$D = (\mathcal{G}, g, f, \nu, g^{c_1}, f^{c_2}), \quad T_1 = \nu^{c_1+c_2}, \quad T_2 \xleftarrow{R} G$$

We define an advantage of an algorithm $\mathcal{A}$ in breaking $DLIN$ to be:

$$Adv_{\mathcal{G},\mathcal{A}}^{DLIN}(\lambda) := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$$

Decisional Bilinear Diffie-Hellman Assumption (*DBDH*) Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (p, G, G_T, e) \xleftarrow{R} \mathcal{G}, \quad g \xleftarrow{R} G, \quad c_1, c_2, c_3 \xleftarrow{R} \mathbb{Z}_p$$

$$D = (\mathcal{G}, g, g^{c_1}, g^{c_2}, g^{c_3}), \quad T_1 = e(g, g)^{c_1 c_2 c_3}, \quad T_2 \xleftarrow{R} G_T$$

We define an advantage of an algorithm $\mathcal{A}$ in breaking $DBDH$ to be:

$$Adv_{\mathcal{G},\mathcal{A}}^{DBDH}(\lambda) := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$$

In our security proofs, we utilize both *DLIN* and *DBDH*. However, it is well known that *DLIN* implies *DBDH* [11, 14].

2.4 Key-Policy Attribute-Based Encryption

KP-ABE system is obtained by four PPT algorithms, Setup, Encrypt, KeyGen and Decrypt. Since we aim to describe a large universe of attributes, the total number of attributes in the universe is not necessary for Setup. Setup only needs the description of the universe of attributes in our definition and a maximum number of attributes per ciphertext, n.

Setup(λ, U, n): The setup algorithm takes as input a security parameter λ, an attribute universe description U and a maximum number of attributes per ciphertext, n. It outputs public parameters PK and a master secret key MSK.

Encrypt(PK, M, S): The encryption algorithm takes as input the public parameters PK, a message M, and a set of attributes S over the universe of attributes.

KeyGen(MSK, PK, $\mathbb{A} = (A, \rho)$): The key generation algorithm takes as input the master secret key MSK, the public parameters PK, and an access structure $\mathbb{A}$. It outputs a private key SK.

Decrypt(PK, CT, SK): The decryption algorithm takes as input the public parameters PK, a ciphertext CT, and a private key SK. If a set of attributes of the ciphertext satisfies an access structure of the private key, it outputs the message M which is encrypted in CT.

2.5 Security Model for KP-ABE

The security model of a semi-adaptive secure KP-ABE is defined as follows.

Setup The challenger runs **Setup** algorithm and sends public parameters PK to the attacker.

Init After all public parameters are published, the attacker chooses a target set of attributes S^* for the challenge ciphertext and gives it to the challenger.

Phase 1 The attacker queries the challenger for private keys corresponding to access structures $\mathbb{A}_1, ..., \mathbb{A}_{q_1}$.

Challenge The attacker sends two messages M_0 and M_1 and a set of attributes S^* such that S^* does not satisfy any of the queried access structures, $\mathbb{A}_1, ..., \mathbb{A}_{q_1}$ to the challenger. The challenger randomly generates $\beta \in \{0, 1\}$, and encrypts M_b under S^*, producing CT^*. It responses to the attacker by sending CT^*.

Phase 2 The attacker queries the challenger for private keys corresponding to sets of access structures $\mathbb{A}_{q_1+1}, ..., \mathbb{A}_q$, with the restriction that none of these are satisfied by S^*. **Guess** The attacker outputs a guess β' for β. The advantage of an attacker in this game is defined to be $\Pr[\beta = \beta'] - 1/2$.

Definition 3 *(semi-adaptively secure KP-ABE) A key policy attribute-based encryption system is semi-adaptively secure if all polynomial time attackers have at most a negligible advantage in the security game above.*

3. KP-ABE WITH SHORT CIPHERTEXTS

We introduce KP-ABE with short ciphertexts in a large universe U. In this system, the maximum number of attributes per ciphertext is bounded by n, but the total number of attributes in the system is not bounded. In this scheme, we achieve short ciphertexts using a formula provided in [7], but leverage an entropy using the nested dual system encryption instead of a q-type assumption. Moreover, our technique achieve additional improvement in the scheme. Our scheme is semi-adaptively secure with a large universe of attributes although the scheme of [7] achieves a selective security with a small universe.

3.1 Construction

$\mathsf{Setup}(\lambda, U, n)$ First, G and $G_T \xleftarrow{R} \mathcal{G}(\lambda, p)$. Then, the algorithm generates $g, v, v_1, v_2, w \in G$ and exponents $a_1, a_2, b, \alpha, h_0, ..., h_n \in \mathbb{Z}_p$. Let $\tau_1 = vv_1^{a_1}, \tau_2 = vv_2^{a_2}$. It publishes the public parameters PK as follows

$$(g, g^b, g^{a_1}, g^{a_2}, g^{b\cdot a_1}, g^{b\cdot a_2}, \tau_1, \tau_2, \tau_1^b, \tau_2^b, w,$$

$$g^{h_0}, ..., g^{h_n}, e(g,g)^{\alpha\cdot a_1\cdot b})$$

The algorithm sets $\text{MSK} := (g^{\alpha}, g^{\alpha\cdot a_1}, v, v_1, v_2)$.

$\mathsf{Encrypt}(\text{PK}, M, S = \{Att_1, ..., Att_u\})$ the algorithm chooses s_1, s_2, t and $cTag$ from $\mathbb{Z}_p$ and sets $s = s_1 + s_2$. It computes $c_0, ..., c_n$ which are coefficients of $y^0, ..., y^n$ in $\prod_{Att_i \in S}(y - Att_i)$, respectively. It outputs a ciphertext CT as follows

$$CT := (C,\, C_1,\, C_2,\, C_3,\, C_4,\, C_5,\, C_6,\, C_7,\, E_0,\, E_1,\, cTag)$$

where

$$C = M \cdot (e(g,g)^{\alpha a_1 \cdot b})^{s_2},\ C_1 = (g^b)^s,\ C_2 = (g^{b\cdot a_1})^{s_1},$$

$$C_3 = (g^{a_1})^{s_1}, C_4 = (g^{b\cdot a_2})^{s_2},\ C_5 = (g^{a_2})^{s_2},$$

$$C_6 = \tau_1^{s_1}\tau_2^{s_2}, C_7 = (\tau_1^b)^{s_1}(\tau_2^b)^{s_2}w^{-t},$$

$$E_0 = g^t,\ E_1 = ((g^{h_0})^{c_0}(g^{h_1})^{c_1}\cdots(g^{h_n})^{c_n}w^{cTag})^t.$$

$\mathsf{KeyGen}(\text{MSK, PK}, \mathbb{A} = (A, \rho))$ We let A_x denote the x^{th} row of A and $\rho(x)$ write an attribute associated A_x by the mapping, ρ. The algorithm randomly chooses $r_{1,x}, r_{2,x}, z_{1,x}, z_{2,x}$ from $\mathbb{Z}_p$ for each $x \in [m]$, and sets $r_x = r_{1,x} + r_{2,x}$ where A is an $m \times \ell$ matrix. It randomly chooses $kTag_{j,x}$ for each $x \in [m]$ and $j \in [n]$ from $\mathbb{Z}_p$. Then, it randomly selects an ℓ size vector $\vec{\mu}$ of which the first coordinate equals to α from $\mathbb{Z}_p^{\ell}$, and sets $\lambda_x = A_x\vec{\mu}$ as the share of A_x. It creates SK as follows

$$SK := (D_{1,x}, D_{2,x}, D_{3,x}, D_{4,x}, D_{5,x}, D_{6,x}, D_{7,x},$$

$$\{K_{j,x}, kTag_{j,x}; \forall j \in [n]\}; \forall x \in [m])$$

where

$$D_{1,x} = g^{\lambda_x \cdot a_1}v^{r_x}, D_{2,x} = g^{-\lambda_x}v_1^{r_x}g^{z_{1,x}},$$

$$D_{3,x} = (g^b)^{-z_{1,x}}, D_{4,x} = v_2^{r_x}g^{z_{2,x}}, D_{5,x} = (g^b)^{-z_{2,x}},$$

$$D_{6,x} = (g^b)^{r_{2,x}},\ D_{7,x} = g^{r_{1,x}},$$

$$K_{j,x} = (g^{h_j}g^{-h_0\rho(x)^j}w^{kTag_{j,x}})^{r_{1,x}} \quad \forall j \in [n].$$

$\mathsf{Decrypt}(SK, CT, PK, \mathbb{A}, S)$ First, the algorithm calculates constants w_x such that $\sum_{\rho(x)\in S}\omega_x A_x = (1, 0, ...0)$. For each $x \in S$, it calculates $c_0, ..., c_n$ by the same manner of $\mathsf{Encrypt}$, then it calculates $Tag_x := \sum_{j=[n]} c_j \cdot kTag_{j,x} - cTag$. If $Tag_x \neq 0$, it calculates

$$W_1 := e(C_1, \prod_{\rho(x)\in S} {D_{1,x}}^{\omega_x}) \cdot e(C_2, \prod_{\rho(x)\in S} {D_{2,x}}^{\omega_x})$$
$$\cdots e(C_5, \prod_{\rho(x)\in S} {D_{5,x}}^{\omega_x})$$

$$W_2 := e(C_6, \prod_{\rho(x)\in S} {D_{6,x}}^{\omega}) \cdot e(C_7, \prod_{\rho(x)\in S} {D_{7,x}}^{\omega})$$

$$W_3 := e(E_1, \prod_{\rho(x)\in S} {D_{7,x}}^{-\omega_x/Tag_x})$$
$$\cdot\, e(E_0, \prod_{\rho(x)\in S, j=[n]} {K_{j,x}}^{c_j\cdot\omega_x/Tag_x}).$$

Otherwise, it aborts. Finally,

$$M = C \cdot \frac{W_2 \cdot W_3}{W_1}.$$

Correctness
For the sake of simplicity, we rewrite

$$\frac{W_1}{W_2 \cdot W_3} = \prod_{\rho(x)\in S}\left(\frac{W_{1,x}}{W_{2,x}\cdot W_{3,x}}\right)^{\omega_x}$$

where $W_{1,x} := e(C_1, D_{1,x}) \cdot e(C_2, D_{2,x}) \cdots e(C_5, D_{5,x})$, $W_{2,x} = e(C_6, D_{6,x}) \cdot e(C_7, D_{7,x})$ and

$$W_{3,x} := \left(\frac{e(E_0, \prod_{j=[n]}(K_{j,x})^{c_j})}{e(E_1, D_{7,x})}\right)^{1/Tag_x}.$$

The computation of $W_{1,x}/W_{2,x}$ is trivial. Also, the similar computations can be appeared in [41]. We only write the result of the computation.

$$W_{1,x}/W_{2,x} = e(g,g)^{\lambda_x a_1 b s_2}e(g,w)^{r_{1,x}t}.$$

To compute $W_{3,x}$, we compute

$$e(E_0, \prod_{j=[n]}(K_{j,x})^{c_j})$$
$$= e(g^t, \prod_{j=[n]}(g^{h_j - h_0\rho(x)^j}w^{kTag_{j,x}})^{c_j\cdot r_{1,x}})$$
$$= e(g, g^{c_1h_1+...+c_nh_n - h_0\sum_{j\in[n]}c_j\rho(x)^j}w^{\sum_{j\in[n]}c_j\cdot kTag_{j,x}})^{t\cdot r_{1,x}}$$
$$= e(g, g^{c_1h_1+...+c_nh_n})^{t\cdot r_{1,x}}e(g, g^{-h_0\sum_{j\in[n]}c_j\rho(x)^j})^{t\cdot r_{1,x}}$$
$$\cdot\, e(g, w^{\sum_{j\in[n]}c_j\cdot kTag_{j,x}})^{t\cdot r_{1,x}}$$
$$= e(g, g^{c_1h_1+...+c_nh_n})^{t\cdot r_{1,x}}e(g,g)^{-t\cdot r_{1,x}\cdot h_0\cdot\sum_{j\in[n]}c_j\rho(x)^j}$$
$$\cdot\, e(g,w)^{t\cdot r_{1,x}\sum_{j\in[n]}c_j\cdot kTag_{j,x}},$$

and

$$
\begin{aligned}
& e(E_1, D_{7,x}) \\
& = e((g^{h_0 c_0 + ... + h_n c_n} w^{cTag})^t, g^{r_{1,x}}) \\
& = e(g^{h_0 c_0 + h_1 c_1 + ... + h_n c_n}, g)^{t \cdot r_{1,x}} e(w, g)^{t \cdot r_{1,x} \cdot cTag}.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
& e(\prod_{j=[n]} (E_0, K_{j,x})^{c_j})/e(E_1, D_{7,x}) \\
& = e(g,g)^{-r_{1,x} \cdot t \cdot h_0 \cdot c_0} e(g,g)^{-r_{1,x} \cdot t \cdot h_0 \sum_{j \in [n]} c_j \rho(x)^j} \\
& \quad \cdot e(g,w)^{r_{1,x} \cdot t((\sum_{j=[n]} c_j \cdot kTag_{j,x}) - cTag)} \\
& = e(g,g)^{-h_0 r_{1,x} \cdot t(c_0 + c_1 \rho(x) + c_2 \rho(x)^2 + ... + c_n \rho(x)^n)} \\
& \quad \cdot e(g,w)^{r_{1,x} \cdot t((\sum_{j=[n]} c_j \cdot kTag_{j,x}) - cTag)} \\
& = e(g,w)^{r_{1,x} \cdot t((\sum_{j=[n]} c_j \cdot kTag_{j,x}) - cTag)}
\end{aligned}
$$

It worth noting that the last equality holds because

$$c_0 + c_1 \rho(x) + c_2 \rho(x)^2 + ... + c_n \rho(x)^n = \prod_{Att_i \in S} (\rho(x) - Att_i) = 0.$$

Because all tags are given,

$$w_{3,x} := \left(\frac{e(E_0, \prod_{j=[n]} (K_{j,x})^{c_j})}{e(C_7 \cdot E_1)} \right)^{1/Tag_x} = e(g,w)^{r_{1,x} \cdot t}$$

Therefore, it can compute

$$
\begin{aligned}
\prod_{\rho(x) \in S} (\frac{W_{1,x}}{W_{2,x} W_{3,x}})^{\omega_x} &= \prod_{\rho(x) \in S} (e(g,g)^{\lambda_x a_1 b s_2})^{\omega_x} \\
&= e(g,g)^{\alpha a_1 b s_2}
\end{aligned}
$$

because $\sum_{\rho(x) \in S} \omega_x \lambda_x = \alpha$. Finally, $M = C/e(g,g)^{\alpha a_1 b s_2}$.

3.2 Security Analysis

We define two semi-functional algorithms SFKeyGen and SFEnc. We remind that since we prove the security under the semi-adaptive security model of KP-ABE, the simulator always knows the target set S^* for the challenge ciphertext when it creates semi-functional keys.

SFKeyGen(MSK, PK, $S^*, \mathbb{A}$) The algorithm takes as inputs the target set of attribute S^* for the challenge ciphertext and an access structure $\mathbb{A} = (A, \rho)$ where A is an $m \times \ell$ matrix. First, to generate the semi-functional key, the algorithm generates a normal key

$$D'_{1,x}, ..., D'_{7,x}, \{K'_{j,x},\ kTag'_{j,x}; \forall j \in [n]\} \qquad \forall x \in [m]$$

using KeyGen. Then, it sets

$$D_{1,x} = D'_{1,x}, ..., D_{7,x} = D'_{7,x},$$

$$\{K_{j,x} = K'_{j,x},\ kTag_{j,x} = kTag'_{j,x}; \forall j \in [n]\} \forall x\ s.t.\ \rho(x) \in S^*$$

For the rest key elements, It randomly selects $\gamma_x, ...\gamma_{x_\theta}$ from $\mathbb{Z}_p$ for each x such that $\rho(x) \notin S^*$ and defines $D_{1,x}$, $D_{2,x}$, $D_{4,x}$ as

$$D_{1,x} = D'_{1,x} \cdot g^{-a_1 a_2 \gamma_x}, D_{2,x} = D'_{2,x} \cdot g^{a_2 \gamma_x},$$

$$D_{4,x} = D'_{4,x} \cdot g^{a_1 \gamma_x} \quad \forall x\ s.t.\ \rho(x) \notin S^*$$

and sets other elements to equal those of the normal key.

SFEncrypt(PK, M, S) For a set of attributes S, the algorithm generates a normal ciphertext

$$C', C'_1, ..., C'_7, E'_0, E'_1, cTag'$$

by using Encrypt. Then, it sets a semi-functional ciphertext identically with the normal ciphertext except $C_4, ..., C_7$. Then, it randomly selects κ from $\mathbb{Z}_p$ and sets $C_4, ..., C_7$ as

$$C_4 = C'_4 \cdot g^{b a_2 \kappa}, C_5 = C'_5 \cdot g^{a_2 \kappa}, C_6 = C'_6 v_2^{a_2 \kappa}, C_7 = C'_7 v_2^{b a_2 \kappa}.$$

In our security proof, we utilize a hybrid model to convert a normal key to a semi-functional key. Instead of changing a type of the key at once, we change the key elements associated with an attribute which is not included in S^* one-by-one. In order to describe this process, we additionally define a semi-functional key generation algorithm SFKeyGen′. It should be noted that the semi-functional key generation algorithm additionally takes as input an index θ.

SFKeyGen′(MSK, PK, $S^*, \mathbb{A}, \theta$) The algorithm takes as inputs an index θ, the target set of attribute S^* for the challenge ciphertext and an access structure $\mathbb{A} = (A, \rho)$ where A is an $m \times \ell$ matrix. We let x_i denote the index of the i^{th} row A_x of A such that $\rho(x) \notin S^*$. To generate the semi-functional key, the algorithm generates a normal key

$$D'_{1,x}, ..., D'_{7,x}, \{K'_{j,x},\ kTag'_{j,x}; \forall j \in [n]\} \qquad \forall x \in [m]$$

using KeyGen.

1. For all x such that $\rho(x) \in S^*$, it sets

$$D_{1,x} = D'_{1,x}, ..., D_{7,x} = D'_{7,x}, \{K_{j,x} = K'_{j,x},$$

$$kTag_{j,x} = kTag'_{j,x}; \forall j \in [n]\} \quad \forall x\ s.t.\ \rho(x) \in S^*$$

2. For all x_i such that $i \leq \theta$, it randomly selects $\gamma_{x_1}, ...\gamma_{x_\theta}$ from $\mathbb{Z}_p$, and defines D_{1,x_i}, D_{2,x_i}, D_{4,x_i} as

$$D_{1,x_i} = D'_{1,x_i} \cdot g^{-a_1 a_2 \gamma_{x_i}}, D_{2,x_i} = D'_{2,x_i} \cdot g^{a_2 \gamma_{x_i}},$$

$$D_{4,x_i} = D'_{4,x_i} \cdot g^{a_1 \gamma_{x_i}}$$

and sets other elements equal to those of the normal key.

3. For all x_i such that $i > \theta$, it sets,

$$D_{1,x_i} = D'_{1,x_i}, ..., D_{7,x_i} = D'_{7,x_i},$$

$$\{K_{j,x_i} = K'_{j,x_i}, kTag_{j,(x_i)} = kTag'_{j,(x_i)}; \forall j \in [n]\}$$

It should be noted that SFKeyGen′(MSK, PK, $S^*, (A, \rho), \Theta$) is identical with SFKeyGen(MSK, PK, $S^*, (A, \rho)$) where Θ is the total number of rows, A_x, such that $\rho(x) \notin S^*$ where A_x is the x^{th} row of the access matrix A.

Game$_{Real}$ This game is identical with the semi-adaptive security model. It should be noted that all keys and the challenge ciphertext are normal in this game.

Game$_{\delta,0}$ is identical with Game$_{\delta-1,\Theta_{\delta-1}}$ where $\Theta_{\delta-1}$ is the total number of rows, A_x, such that $\rho(x) \notin S^*$ of the $\delta - 1^{th}$ key. In this game, the first $\delta - 1$ keys are generated by SFKeyGen(MSK, PK, $S^*, \mathbb{A}$). It should be noted that in Game$_{0,0}$ all keys are normal, but the challenge ciphertext is semi-functional.

Game$_{\delta,\theta}$ We let x_i denote the index of the i^{th} row A_x of A such that $\rho(x) \notin S^*$ where (A, ρ) is an access structure for the δ^{th} key. This game is identical with $\mathsf{Game}_{\delta,\theta-1}$ except the key elements for A_{x_θ} of the δ^{th} key. In this game, the key elements for $A_{x_1}, ..., A_{x_\theta}$ are semi-functional. It means $\mathsf{SFKeyGen}'$(MSK, PK, $S^*, \mathbb{A}, \theta$) are used to generate the δ^{th} key.

Game_{Final} This game is identical with $\mathsf{Game}_{q,\Theta_q}$ except the message encrypted in the challenge ciphertext where q is the total number of key queries in Phase I and Phase II. In this game, a random message replaces the message in the challenge ciphertext.

THEOREM 1. *Our KP-ABE scheme with short ciphertexts is semi-adaptively secure under the decisional linear assumption.*

Proof: This is proved by Lemmas 1, 2, and 3. □

Lemma 1. (Semi-functional ciphertext invariance) *Suppose there exists a PPT algorithm $\mathcal{A}$ to distinguish between $Game_{Real}$ and $Game_{0,0}$ with a non-negligible advantage ϵ. Then we can build an algorithm $\mathcal{B}$ breaking DLIN with the advantage, ϵ, using $\mathcal{A}$.*

Proof: This proof is similar with the proof of Lemma 1. $\mathcal{B}$ takes $(g, f, \nu, g^{c_1}, f^{c_2}, T)$ as an instance from $DLIN$ assumption. It will simulate either $Game_{Real}$ or $Game_{0,0}$ based on the value of T.

Setup: The algorithm selects $a_1, b, y_v, y_{v_1}, y_{v_2}, y_w, h_0, ..., h_n$ from $\mathbb{Z}_p$, and sets $g^{a_1} = f, g^{a_2} = \nu$. Then, it publishes the public parameters as follows

$$g,\ g^b,\ g^{b\cdot a_1} = f^b,\ g^{b\cdot a_2} = (\nu)^b,\ w = g^{y_w},\ g^{h_0},\ ...,\ g^{h_n},$$

$$\tau_1 = f^{y_{v_1}},\ \tau_2 = \nu^{y_{v_2}},\ \tau_1^b, \tau_2^b,\ e(g,g)^{\alpha\cdot a_1\cdot b} = e(g,f)^{\alpha\cdot b}.$$

It also sets MSK $= \{g^\alpha, g^{\alpha\cdot a_1} = f^\alpha, v, v_1, v_2\}$.

Init: Before it generates any private key, $\mathcal{B}$ requests to the adversary a target set of attributes S^* which will be used to generate the challenge ciphertext.

Phase I and II: To generate normal keys, $\mathcal{B}$ uses the key generation algorithm, KeyGen. It is possible because $\mathcal{B}$ knows all public parameter and MSK.

Challenge: When $\mathcal{A}$ requests the challenge ciphertext for $S^* = \{Att_1, ..., Att_u\}$ with two message M_0 and M_1, $\mathcal{B}$ randomly selects β from $\{0, 1\}$. Then, it generates a normal ciphertext, $C', C'_1, ..., C'_7, E'_0, E'_2, cTag'$ using **Encrypt**. It sets the challenge ciphertext as follows:

$$C = C' \cdot \left(e(g^{c_1}, f) \cdot e(g, f^{c_2})\right)^{b\cdot\alpha},$$

$$C_1 = C'_1, (g^{c_1})^b, C_2 = C'_2 \cdot \left(f^{c_2}\right)^{-b},\ C_3 = C'_3 \cdot (f^{c_2}),$$

$$C_4 = C'_4(T)^b,\ C_5 = C'_5 {\cdot} T,\ C_6 = C'_6 {\cdot} (g^{c_1})^{y_v} {\cdot} (f^{c_2})^{-y_{v_1}} {\cdot} T^{y_{v_2}},$$

$$C_7 = C'_7 \cdot \left((g^{c_1})^{y_v} \cdot (f^{c_2})^{-y_{v_1}} \cdot T^{y_{v_2}}\right)^b,$$

$$E_0 = E'_0,\ E_1 = E'_1,\ cTag = cTag'$$

We let s'_1, s'_2, t denote the randomization parameters of the normal challenge ciphertext. This implicitly sets $s_1 = -c_2 + s'_1$ and $s_2 = s'_2 + c_1 + c_2$. Therefore, if T equals to $\nu^{c_1+c_2}$, $\mathcal{B}$ has properly simulated $Game_{Real}$. Otherwise, if T is a random value, it has simulated $Game_{0,0}$, also properly by letting $\nu^{c_1+c_2} g^\kappa$ denote T. □

To achieve *semi-functional key invariance*, we designed *a nested duals system encryption*. In our proof of lemma 2, the independence of *tag*s in the challenge key and the challenge ciphertext is proved by n-wise independence [6]. In detail, *tag*s for A_x, $kTag_{j,x}\, \forall j \in [n]$ and a *tag* for the challenge ciphertext, $cTag$ are generated as

$$\begin{pmatrix} -\rho(x) & 1 & & & \\ -(\rho(x))^2 & & 1 & & \\ \vdots & & & \ddots & \\ -(\rho(x))^n & & & & 1 \\ c_0 & c_1 & c_2 & \cdots & c_n \end{pmatrix} \begin{pmatrix} h'_0 \\ h'_1 \\ h'_2 \\ \vdots \\ h'_n \end{pmatrix} = \begin{pmatrix} kTag_{1,x} \\ kTag_{2,x} \\ \vdots \\ kTag_{n,x} \\ cTag \end{pmatrix}$$

where c_j is coefficients of y^j of $\prod_{\rho(x)\in S^*}(y - \rho(x))$ and S^* is the target set of attributes for the challenge ciphertext.

To claim n-wise independence of *tag*s, we show that they satisfy two conditions following

1. $h'_0, ...h'_n$ are information theoretically hidden to the adversary.

2. $\rho(x)$ is not in S^*.

In lemma 2, we must show that they appear only once to suffice key elements for A_x for the first condition. To do this, we isolates $kTag_{j,x}\, \forall j \in [n]$ for A_x from the other tags by utilizing a hybrid model. In the security proof, we show the invariance of two games which have different types of key elements corresponding to A_x, not all key elements in the k^{th} key. Also, we do not apply semi-functionality for the elements if a corresponding attribute of A_x is shared between keys and the challenge ciphertext. Therefore, we leave those key elements as normal type (i.e. without any change of type from the real game). This is because the correlation can be detected to the adversary by checking $\sum_{i\in[0,n]} c_i kTag_{i,\rho(x)} = cTag$ if $\rho(x)$ is in S^* as the second condition requires.

Lemma 2. (Semi-functional key invariance) *Suppose there exists a PPT algorithm $\mathcal{A}$ to distinguish between $Game_{k,\theta-1}$ and $Game_{k,\theta}$ with a non-negligible advantage ϵ. Then we can build an algorithm $\mathcal{B}$ breaking DLIN with the advantage, ϵ, using $\mathcal{A}$.*

Proof: First, $\mathcal{B}$ takes $(g, f, \nu, g^{c_1}, f^{c_2}, T)$ as an instance from $DLIN$. It will simulate either $Game_{k,\theta-1}$ or $Game_{k,\theta}$ based on the value of T.

Setup: The algorithm selects $\alpha, a_1, a_2, y_{v_1}, y_{v_2}, y_w, h'_0, ..., h'_n$, $\tilde{h}_0, ..., \tilde{h}_n$ from $\mathbb{Z}_p$, and sets

$$g = g, g^b = f, v = \nu^{-a_1\cdot a_2}, v_1 = \nu^{a_2} \cdot g^{y_{v_1}}, v_2 = \nu^{a_1} \cdot g^{y_{v_2}},$$

$$w = fg^{y_w}, h_0 = f^{-h'_0} g^{\tilde{h}_0}, ..., h_n = f^{-h'_n} g^{\tilde{h}_n}$$

We do not know h_i, but we can calculate h_i using g, f, h'_i, and $\tilde{h}_i$. It publishes the public parameter following

$$g, g^b, g^{b\cdot a_1} = f^{a_1}, g^{b\cdot a_2} = f^{a_2}, g^{h_0}, ..., g^{h_n}, w,$$

$$\tau_1 = g^{y_{v_1} a_1}, \tau_2 = g^{y_{v_2} a_2}, \tau_1^b = f^{y_{v_1} a_1}, \tau_2^b = f^{y_{v_2} a_2},$$

$$e(g,g)^{\alpha\cdot a_1 b} = e(f,g)^{\alpha\cdot a_1}.$$

$\mathcal{B}$ generates MSK$=$ $\{g^\alpha, g^{\alpha\cdot a_1}, v, v_1, v_2\}$. This is possible because it knows a_1 and α. We stress that $\mathcal{B}$ does not require any information of the target set of attributes, S^* for the challenge ciphertext when it sets all parameters in Setup.

Table 2: Summary of Security Games

	Key Generation Algorithm	Encryption Algorithm
Game_{Real}	KeyGen	Enc
$\mathsf{Game}_{0,0}$	KeyGen	SFEnc
$\mathsf{Game}_{\delta,0}$	SFKeyGen $(<\delta)$ KeyGen $(\geq\delta)$	SFEnc
$\mathsf{Game}_{\delta,\theta}$	SFKeyGen $(<\delta)$ SFKeyGen′ with θ $(=\delta)$ KeyGen $(>\delta)$	SFEnc
$\mathsf{Game}_{q,\Theta_q}$	SFKeyGen	SFEnc
Game_{Final}	SFKeyGen	SFEnc with a random message

$(<\delta)$: For the first $\delta-1$ keys, $(=\delta)$: For the δ^{th} key,
$(>\delta)$: For all keys except the first δ keys

Init: Before it generates any private key, $\mathcal{B}$ requests to the adversary a target set of attributes S^* which will be used to generate the challenge ciphertext.

Phase I and II: For the first $k-1$ keys $(<k)$, first it generates a normal key $(D'_{1,x},...,D'_{7,x}, \{K'_{j,x}, kTag'_{j,x};\forall j\in[n]\};\forall x\in[m])$ where m is the number of rows of an access matrix for the key, then, it selects γ_x from $\mathbb{Z}_p$ for each x such that $\rho(x)\notin S^*$. For all rows of A, it sets

1. $\forall x\ s.t.\ \rho(x)\in S^*$

$$D_{1,x}=D'_{1,x}, D_{2,x}=D'_{2,x}, D_{3,x}=D'_{3,x}D_{4,x}=D'_{4,x},$$

$$D_{5,x}=D'_{5,x}, D_{6,x}=D'_{6,x}, D_{7,x}=D'_{7,x},$$

$$\{K_{j,x}=K'_{j,x}, kTag_{j,x}=kTag'_{j,x};\forall j\in[n]\}).$$

2. $\forall x\ s.t.\ \rho(x)\notin S^*$

$$D_{1,x}=D'_{1,x}\cdot g^{-a_1a_2\gamma_x}, D_{2,x}=D'_{2,x}\cdot g^{a_2\gamma_x}, D_{3,x}=D'_{3,x},$$

$$D_{4,x}=D'_{4,x}\cdot g^{a_1\gamma_x}, D_{5,x}=D'_{5,x}, D_{6,x}=D'_{6,x},$$

$$D_{7,x}=D'_{7,x},\ \{K_{j,x}=K'_{j,x}, kTag_{j,x}=kTag'_{j,x};\forall j\in[n]\}.$$

For the rest keys except the k^{th} key $(>k)$, $\mathcal{B}$ runs the key generation algorithm to generate normal keys. It is possible since $\mathcal{B}$ knows all public parameters and MSK.

We let x_i denote the index of the i^{th} A_x of A such that $\rho(x)\notin S^*$. To generate the k^{th} key with the index θ, for $\mathbb{A}=(A,\rho)$, it first generates $\{kTag_{j,x};\forall j\in[n]\}$ randomly from $\mathbb{Z}_p$ for each $x\ s.t.\ x\neq x_\theta$. For x_θ, it sets

$$kTag_{j,x_\theta}=h'_j+h'_0\rho(x_\theta)^j\quad\forall j\in[n].$$

Then, using $\{kTag_{j,x};\forall j\in[n],\forall x\in[m]\}$, it generates a normal key

$$D'_{1,x},...,D'_{7,x},\{K'_{j,x}, kTag_{j,x};\forall j\in[n]\}\quad\forall x\in[m].$$

Finally, it selects γ_{x_i} from $\mathbb{Z}_p$ for each x_i such that $i<\theta$. To set the k^{th} key, for all x such that $\rho(x)\in S^*$, it sets

$$D_{1,x}=D'_{1,x},...,D_{7,x}=D'_{7,x},$$

$$\{K_{j,x}=K'_{j,x}, kTag_{j,x};\forall j\in[n]\}.$$

For the rest of the key (i.e. $\forall x\ s.t.\ \rho(x)\notin S^*$), it sets as follows:

$\forall x_i\ s.t.\ i<\theta,$

$$D_{1,x_i}=D'_{1,x_i}\cdot g^{-a_1a_2\gamma_{x_i}},\ D_{2,x_i}=D'_{2,x_i}\cdot g^{a_2\gamma_{x_i}},$$

$$D_{3,x_i}=D'_{3,x_i}, D_{4,x_i}=D'_{4,x_i}\cdot g^{a_1\gamma_{x_i}},\ D_{5,x_i}=D'_{5,x_i},$$

$$D_{6,x_i}=D'_{6,x_i}, D_{7,x_i}=D'_{7,x_i},$$

$$\{K_{j,x_i}=K'_{j,x_i},\ kTag_{j,x_i};\forall j\in[n]\}.$$

For x_θ (i.e. $i=\theta$),

$$D_{1,x_\theta}=D'_{1,x_\theta}T^{-a_1a_2},\ D_{2,x_\theta}=D'_{2,x_\theta}T^{a_2}(g^{c_1})^{y_{v_1}},$$

$$D_{3,x_\theta}=D'_{3,x_\theta}(f^{c_2})^{y_{v_1}}, D_{4,x_\theta}=D'_{4,x_\theta}T^{a_1}(g^{c_1})^{y_{v_2}},$$

$$D_{5,x_\theta}=D'_{5,x_\theta}(f^{c_2})^{y_{v_2}}, D_{6,x_\theta}=D'_{6,x_\theta}f^{c_2},$$

$$D_{7,x_\theta}=D'_{7,x_\theta}(g^{c_1}),$$

$$\{K_{j,x_\theta}=K'_{j,x_\theta}(g^{c_1})^{\tilde{h}_j-\tilde{h}_0\cdot\rho(x_\theta)^j}, kTag_{j,x_\theta};\forall j\in[n]\}.$$

$\forall x_i\ s.t.\ i>\theta,$

$$D_{1,x_i}=D'_{1,x_i},...,D_{7,x_i}=D'_{7,x_i},$$

$$\{K_{j,x_i}=K'_{j,x_i}, kTag_{j,x_i};\forall j\in[n]\}$$

We let $z'_{1,x_\theta}, z'_{2,x_\theta}, r'_{1,x_\theta}, r'_{2,x_\theta}, \mu'$ denote randomized parameters for A_{x_θ} in the normal key. This implicitly sets $z_{1,x_\theta}=z'_{1,x_\theta}-y_{v_1}c_2$ and $z_{2,x_\theta}=z'_{2,x_\theta}-y_{v_2}c_2$. Also, it sets $r_{2,x_\theta}=r'_{2,x_\theta}+c_2$ and $r_{1,x_\theta}=r'_{1,x_\theta}+c_1$. If T equals to $\nu^{c_1+c_2}$, this has properly simulated the semi-functional key generated by SFKeyGen′(MSK, PK, $S^*,\mathbb{A},\theta-1$). Otherwise, If T is a random value, and we denote it as $\nu^{c_1+c_2}g^{\gamma_{x_\theta}}$, this is properly simulated the semi-functional key generated by SFKeyGen′(MSK, PK, $S^*,\mathbb{A},\theta$).

Challenge: When $\mathcal{A}$ requests the challenge ciphertext for S^* with two message, M_0 and M_1 $\mathcal{B}$ randomly selects β from $\{0,1\}$. Then, it sets $cTag=c_0h'_0+...+c_nh'_n$. With M_β and $cTag$, it generates a normal challenge ciphertext, $(C',C'_1,...,C'_7,E_0,E_{Att_i};\forall Att_i\in S^*)$. Then, it randomly generates κ from $\mathbb{Z}_p$ and sets

$$C=C', C_1=C'_1, C_2=C'_2, C_3=C'_3, C_4=C'_4\cdot f^{a_2\kappa},$$

$$C_5=C'_5\cdot g^{a_2\kappa}, C_6=C'_6\cdot v_2^{a_2\kappa},$$

$$C_7 = C_7' \cdot f^{y_{v_2} \cdot \kappa \cdot a_2} \nu^{-a_1 \cdot \kappa \cdot y_w \cdot a_2}, E_0 = E_0' \cdot \nu^{a_1 a_2 \kappa},$$

$$E_1 = E_1' \cdot (\nu^{c_0 \tilde{h}_0 + \ldots + c_n \tilde{h}_n + y_w \cdot cTag})^{a_1 a_2 \kappa}, cTag.$$

This implicitly sets $g^t = g^{t'} \cdot \nu^{a_1 a_2 \kappa}$ where t' is a randomization parameter of a normal key. It should be noted that *tag*s in the challenge ciphertext and the key elements of A_{x_θ} of the k^{th} key are not correlated because of n-wise independence. Hence, all *tag*s in the challenge ciphertext and the k^{th} key are randomly distributed and do not correlate to each other. Therefore, if T equals to $\nu^{c_1+c_2}$, $\mathcal{B}$ has properly simulated $Game_{k,\theta-1}$. Otherwise, it has simulated $Game_{k,\theta}$, also properly. □

Lemma 3. **(Semi-functional Security)** *Suppose there exists a PPT algorithm $\mathcal{A}$ to distinguish between $Game_{q,\Theta_q}$ and $Game_{Final}$ with a non-negligible advantage ϵ. Then, we can build an algorithm $\mathcal{B}$ breaking DBDH with the advantage, ϵ, using $\mathcal{A}$.*

Proof: $\mathcal{B}$ takes $(g, g^{c_1}, g^{c_2}, g^{c_3}, T)$ as an instance from $DBDH$. It will simulate either $Game_{q,\Theta_q}$ or $Game_{Final}$ based on the value of T.

Setup: The algorithm selects $a_1, b, y_v, y_{v_1}, y_{v_2}, y_w,$ $h_0, ..., h_n$ from $\mathbb{Z}_p$, and sets

$$g^{a_2} = g^{c_2}, v = g^{y_v}, v_1 = g^{y_{v_1}}, v_2 = g^{y_{v_2}}, w = g^{y_w},$$

$$\{g^{h_i}; i \in [0, n]\}$$

Then, it publishes the public parameters as follows

$$g, g^b, g^{b \cdot a_1}, g^{b \cdot a_2} = (g^{c_2})^b, w, g^{h_0}, ..., g^{h_n},$$

$$\tau_1 = {v_1}^{a_1}, \tau_2 = (g^{c_2})^{y_{v_2}}, \tau_1^b, \tau_2^b, e(g,g)^{\alpha \cdot a_1 b} = e(g^{c_1}, g^{c_2})^{a_1 b}.$$

This implicitly sets $\alpha = c_1 \cdot c_2$ and $a_2 = c_2$. In this setting, the simulator does not know MSK since it does not know α.

Init: Before it generates any private key, $\mathcal{B}$ requests to the adversary a target set of attributes S^* which will be used to generate the challenge ciphertext.

Phase I and II: For generating a semi-functional key, $\mathcal{B}$ randomly selects $\vec{\mu}_1$ from $\mathbb{Z}_p$ such that $A_x \cdot \vec{\mu}_1 = 0$ for all x such that $\rho(x) \in S^*$ and the first coordinate of $\vec{\mu}_1$ equals to 1. This exists because S^* does not satisfy an access structure of the semi-functional key. Also, it generate a random vector $\vec{\mu}_2$ of which the first coordinate equals to 0. It implicitly sets $\alpha \cdot \vec{\mu} = \alpha \cdot \vec{\mu}_1 + \vec{\mu}_2$. For each $x \in [m]$, it randomly generates $z_{1,x}, z_{2,x}, r_{1,x}, r_{2,x},$ $\{kTag_{j,x}; \forall j \in [n]\}$ from $\mathbb{Z}_p$.

Then, for normal type rows (i.e. $\forall x \ s.t. \ \rho(x) \in S^*$), it sets

$$D_{1,x} = g^{A_x \mu_2 \cdot a_1} v^{r_x}, D_{2,x} = g^{-A_x \mu_2} v_1^{r_x} g^{z_{1,x}},$$

$$D_{3,x} = (g^b)^{-z_{1,x}}, D_{4,x} = v_2^{r_x} g^{z_{2,x}}, D_{5,x} = (g^b)^{-z_{2,x}},$$

$$D_{6,x} = g^{r_{2,x} \cdot b}, D_{7,x} = g^{r_{1,x}},$$

$$\{K_{j,x} = (h_j h_0^{-\rho(x)^j} w^{kTag_{j,x}})^{r_{1,x}}, kTag_{j,x}; j \in [n]\}$$

For the rest semi-functional rows, it randomly generate γ_x' for each x such that $\rho(x) \notin S^*$. It sets

$$D_{1,x} = g^{A_x \mu_2 \cdot a_1} (g^{c_2})^{-\gamma_x' \cdot a_1} v^{r_x},$$

$$D_{2,x} = g^{-A_x \mu_2} (g^{c_2})^{-\gamma_x'} v_1^{r_x} g^{z_{1,x}}, D_{3,x} = (g^b)^{-z_{1,x}},$$

Table 3: The Size of PK, SK and CT (bits)

	Param	AC16 [3]	Ours	Ours/AC16
PK	d.159	8,586	6,360	0.74
	d.201	10,854	8,040	
	d.224	12,096	8,960	
SK	d.159	66,780	45,812	0.69
	d.201	84,420	57,908	
	d.224	94,080	64,532	
CT	d.159	4,770	3,975	0.83
	d.1201	6,030	5,025	
	d.224	6,720	5,600	

$$D_{4,x} = v_2^{r_x} g^{z_{2,x}} (g^{c_1})^{a_1 A_x \mu_1} g^{a_1 \cdot \gamma_x'}, D_{5,x} = (g^b)^{-z_{2,x}},$$

$$D_{6,x} = g^{r_{2,x} \cdot b}, D_{7,x} = g^{r_{1,x}},$$

$$\{K_{j,x} = (h_j h_0^{-\rho(x)^j} w^{kTag_{j,x}})^{r_{1,x}}, kTag_{j,x}; j \in [n]\}$$

This implicitly sets $\gamma_x = c_1 A_x \mu_1 + \gamma_x'$.

Challenge: When $\mathcal{A}$ requests the challenge ciphertext for S^* with two message M_0 and M_1, first, $\mathcal{B}$ randomly selects β from $\{0, 1\}$. It, then, randomly generates s_1, κ and t from $\mathbb{Z}_p$ and sets $s_2 = c_3$ and $\kappa = -c_3 + \kappa'$. It, also, sets the challenge ciphertext as follow.

$$C = M_\beta T^{a_1 b}, C_1 = g^{b s_1} (g^{c_3})^b, C_2 = g^{b a_1 s_1}, C_3 = g^{a_1 s_1},$$

$$C_4 = (g^{c_2})^{b \cdot \kappa'}, C_5 = (g^{c_2})^{\kappa'}, C_6 = \tau_1^{s_1} (g^{c_3})^{y_v} (g^{c_2})^{y_{v_2} \cdot \kappa'},$$

$$C_7 = \tau_1^{s_1 \cdot b} (g^{c_3})^{y_v \cdot b} (g^{c_2})^{y_{v_2} \cdot \kappa' b} w^{-t},$$

$$E_0 = g^t, E_1 = (h_0^{c_0} h_1^{c_1} ... h_n^{c_n} w^{cTag})^t$$

If T equals to $g^{c_1 c_2 c_3}$, $\mathcal{B}$ has properly simulated $Game_{q,\Theta_q}$. Otherwise, a random will be added in M_β, and it has simulated $Game_{Final}$, also properly. □

4. IMPLEMENTATION AND BENCHMARK

We implement our scheme using PBC library [30]. We mainly compare our scheme to Agrawal and Chase's scheme [3]. To compare the efficiency of our scheme, we convert our scheme to asymmetric groups (Appendix 1) since Agrawal and Chase's construction is based only on asymmetric groups. It should be noted that our scheme naturally extended to asymmetric groups since it is constructed in symmetric groups, but the other way around does not trivially hold. Our implementation is executed on VirtualBox [1]. We allocate 4GB memory and 2 CPUs (Intel® Core™i7-4600U CPU @ 2.10GHz x 2) to Ubuntu 16.04 LTS (64 bits).

We evaluate our scheme with the elliptic curves introduced by Miyaji, Nakabayashi and Takano (MNT) with various field sizes (159, 201 and 224 bits) [31]. We are using a simple policy "(A AND B) OR (E OR F)" for the ciphertext. A user has four attributes ("A", "B", "C", "D"). Therefore, it can decrypt the ciphertext using attributes "A" and "B". The maximum number of attributes per a ciphertext is set as five (n=5). We ignore the time to compute LSSS for our policy. It is precomputed outside the implementation. The time to compute LSSS also may be negligible in a real application since the other operations such as pairings and exponentiations over groups are much slower.

Table 4: Comparison of pairing and exponentiation

	AC16 [3]	Ours
Setup	$2(n+6)(E_1+E_2)+P$	$(11+n)E_1+(9+n)E_2+P$
KeyGen	$(4nk+10k+8)E_2$	$(3n+13)kE_2$
Encrypt	$(2n+18)E_1+E_T$	$(n+13)E_1+E_T$
Decrypt	$(6+6m)P+2mnE_1$ $+2mE_T$	$9P+(n+8)mE_2$

n: the maximum number of attributes per a ciphertext.
m: the number of attributes used for decryption.
k: the number of attributes that a user has.
E_1: Exponentiation over G_1, E_2: Exponentiation over G_2
E_T: Exponentiation over G_T
P: Pairing $e : G_1 \times G_2 \rightarrow G_T$

Table 5: The Excution Time (ms)

	Param	AC16 [3]	Ours	Ours/AC16
Setup	d.159	266.5	91.6	0.34
	d.201	344.1	118	0.34
	d.224	439.4	145.3	0.33
KeyGen	d.159	407.4	416.2	1.02
	d.201	526.7	538.2	1.02
	d.224	671.2	670.6	1.00
Encrypt	d.159	16.8	10.6	0.63
	d.201	22.9	14.5	0.63
	d.224	29.7	18.1	0.61
Decrypt	d.159	105.6	89.6	0.85
	d.201	137.3	117.7	0.86
	d.224	176.5.7	145	0.82

The efficiency gap between ours and Agrawal and Chase's scheme increases as the maximum number of attributes per a ciphertext n increases. Our scheme can save up 50 percent of PK and 41 percent of SK if n is big enough. However, even if we set n to be small, parameters of our scheme are much shorter than those of Agrawal and Chase's scheme. Table 3 compares the sizes of keys and ciphertexts between our scheme and Agrawal and Chase's scheme when n is only five. The column *Param* means the field sizes of MNT.

We provide Tables 4 and 5 to show the theoretical and experimental improvements of our scheme. Table 4 shows the number of operations to run Setup, KeyGen, Encrypt and Decrypt of our scheme is significantly less than those of Agrawal and Chase's scheme. We additionally provide the average execution times of those algorithms in Table 5. On running each algorithm 100 times, our scheme is faster Agrawal and Chase's scheme. In particular, compared with Agrawal and Chase's scheme, the Encrypt and Decrypt of our scheme is reduced the execution times more than 35 percent and 14 percent, respectively. The time for Setup of our scheme is much faster than that of Agrawal and Chase's scheme. It only takes 34 percent of their scheme.

5. CONCLUSION

In this paper, we provided a semi-adaptively secure KP-ABE scheme in prime order groups. We achieved an efficient KP-ABE scheme with short ciphertexts. Our KP-ABE with short ciphertexts shows additional desirable properties such as supporting a large attribute universe and multi-use of attributes under the standard assumption. In addition, it has more efficient features than the previous best scheme from both time and space perspectives. We leave a question on how to achieve all those properties in an adaptively secure KP-ABE as an open problem.

Acknowledgments

Jongkil Kim is supported by Australia-India Strategic Research Fund (AISRF).

6. REFERENCES

[1] VirtualBox. https://www.virtualbox.org/. Accessed: 2016-11-04.

[2] M. Abdalla and R. D. Prisco, editors. *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, volume 8642 of *Lecture Notes in Computer Science*. Springer, 2014.

[3] S. Agrawal and M. Chase. A study of pair encodings: Predicate encryption in prime order groups. In E. Kushilevitz and T. Malkin, editors, *TCC*, volume 9563 of *LNCS*, pages 259–288. Springer, 2016.

[4] N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577. Springer, 2014.

[5] N. Attrapadung. Dual system encryption framework in prime-order groups. *IACR Cryptology ePrint Archive*, 2015:390, 2015.

[6] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 384–402. Springer, 2010.

[7] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Catalano et al. [12], pages 90–108.

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

[9] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[10] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In Cramer [17], pages 440–456.

[11] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.

[12] D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors. *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9,*

2011. Proceedings, volume 6571 of *Lecture Notes in Computer Science*. Springer, 2011.

[13] J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 435–460. Springer, 2013.

[14] J. Chen and H. Wee. Doubly spatial encryption from DBDH. *Theor. Comput. Sci.*, 543:79–89, 2014.

[15] J. Chen and H. Wee. Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In Abdalla and Prisco [2], pages 277–297.

[16] J. H. Cheon. Security analysis of the strong diffie-hellman problem. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2006.

[17] R. Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.

[18] C. Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In K. Kurosawa, editor, *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2007.

[19] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Gilbert [21], pages 44–61.

[20] C. Gentry and B. Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[21] H. Gilbert, editor. *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*. Springer, 2010.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[23] A. Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In M. J. J. Jr., M. E. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *ACNS*, volume 7954 of *Lecture Notes in Computer Science*, pages 357–372. Springer, 2013.

[24] J. Kim and S. Nepal. A cryptographically enforced access control with a flexible user revocation on untrusted cloud storage. *Data Science and Engineering*, 1(3):149–160, 2016.

[25] J. Kim, W. Susilo, F. Guo, and M. H. Au. A tag based encoding: An efficient encoding for predicate encryption in prime order groups. In V. Zikas and R. D. Prisco, editors, *SCN*, volume 9841 of *LNCS*, pages 3–22, 2016.

[26] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.

[27] 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 Gilbert [21], pages 62–91.

[28] A. B. Lewko and B. Waters. Unbounded hibe and attribute-based encryption. In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567. Springer, 2011.

[29] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 180–198. Springer, 2012.

[30] B. Lynn. *On the implementation of pairing-based cryptosystems*. PhD thesis, PhD thesis, Stanford Univeristy, 2007.

[31] A. Miyaji, M. Nakabayashi, and S. Takano. Characterization of elliptic curve traces under fr-reduction. In D. Won, editor, *ICISC*, volume 2015 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2000.

[32] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 214–231. Springer, 2009.

[33] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208. Springer, 2010.

[34] T. Okamoto and K. Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In D. Lin, G. Tsudik, and X. Wang, editors, *Cryptology and Network Security - 10th International Conference, CANS 2011, Sanya, China, December 10-12, 2011. Proceedings*, volume 7092 of *Lecture Notes in Computer Science*, pages 138–159. Springer, 2011.

[35] T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In X. Wang and K. Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366. Springer, 2012.

[36] Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM Conference on Computer and Communications Security*, pages 463–474. ACM, 2013.

[37] A. Sahai and B. Waters. Fuzzy identity-based encryption. In Cramer [17], pages 457–473.

[38] Y. Sakemi, G. Hanaoka, T. Izu, M. Takenaka, and M. Yasuda. Solving a discrete logarithm problem with auxiliary input on a 160-bit elliptic curve. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*, pages 595–608. Springer, 2012.

[39] K. Takashima. Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption. In Abdalla and Prisco [2], pages 298–317.

[40] A. Tassanaviboon and G. Gong. Oauth and abe based authorization in semi-trusted cloud computing: Aauth. In *Proceedings of the Second International Workshop on Data Intensive Computing in the Clouds*, DataCloud-SC '11, pages 41–50. ACM, 2011.

[41] B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.

[42] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Catalano et al. [12], pages 53–70.

[43] Z. Xu and K. M. Martin. Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage. In G. Min, Y. Wu, L. C. Liu, X. Jin, S. A. Jarvis, and A. Y. Al-Dubai, editors, *TrustCom*, pages 844–849. IEEE Computer Society, 2012.

APPENDIX

A. KP-ABE WITH SHORT CIPHERTEXTS

We introduce KP-ABE with short ciphertexts with an asymmetric pairing. This scheme is the natural extension of the scheme in Section 3. We use subscript to denote where group elements belong to (e.g $g_1 \in G_1$ and $g_2 \in G_2$).

A.1 Construction

Setup(λ, U, n) First, G_1, G_2 and $G_T \overset{R}{\leftarrow} \mathcal{G}(\lambda, p)$. Then, the algorithm generates $g_1 \in G_1, g_2 \in G_2$ and exponents $y_v, y_w, y_{v1}, y_{v2}, a_1, a_2, b, \alpha, h_0, ..., h_n \in \mathbb{Z}_p$. Let $\tau_1 = y_v + a_1 y_{v1}, \tau_2 = y_v + a_2 y_{v2}$. It publishes the public parameters PK as follows

$$(g_1, g_1^b, g_1^{a_1}, g_1^{a_2}, g_1^{b\cdot a_1}, g_1^{b\cdot a_2}, g_1^{\tau_1}, g_1^{\tau_2}, g_1^{b\tau_1}, g_1^{b\tau_2}, w_1 = g_1^{y_w},$$

$$g_1^{h_0}, ..., g_1^{h_n}, e(g_1, g_2)^{\alpha\cdot a_1\cdot b})$$

The algorithm sets MSK as follows

$$(g_2, g_2^{\alpha}, g_2^{a_1}, g_2^{\alpha\cdot a_1}, g_2^b, v_2 = g_2^{y_v}, u_2 = g_2^{y_{v1}}, u_2' = g_2^{y_{v2}},$$

$$g_2^{h_0}, ..., g_2^{h_n}, w_2 = g_2^{y_w})$$

Encrypt(PK, M, $S = \{Att_1, ..., Att_u\}$) the algorithm chooses s_1, s_2, t and $cTag$ from $\mathbb{Z}_p$ and sets $s = s_1 + s_2$. It computes $c_0, ..., c_n$ which are coefficients of $y^0, ..., y^n$ in $\prod_{Att_i \in S}(y - Att_i)$, respectively. It outputs a ciphertext CT as follows

$$CT := (C,\, C_1,\, C_2,\, C_3,\, C_4,\, C_5,\, C_6,\, C_7,\, E_0,\, E_1,\, cTag)$$

where

$$C = M \cdot (e(g_1, g_2)^{\alpha a_1 \cdot b})^{s_2},\, C_1 = (g_1^b)^s,\, C_2 = (g_1^{b\cdot a_1})^{s_1},$$

$$C_3 = (g_1^{a_1})^{s_1}, C_4 = (g_1^{b\cdot a_2})^{s_2},\, C_5 = (g_1^{a_2})^{s_2},$$

$$C_6 = (g_1^{\tau_1})^{s_1}(g_1^{\tau_2})^{s_2}, C_7 = (g_1^{b\tau_1})^{s_1}(g_1^{b\tau_2})^{s_2} w_1^{-t},$$

$$E_0 = g_1^t,\, E_1 = ((g_1^{h_0})^{c_0}(g_1^{h_1})^{c_1} \cdots (g_1^{h_n})^{c_n} w_1^{cTag})^t.$$

KeyGen(MSK, PK, $\mathbb{A} = (A, \rho)$) We let A_x denote the x^{th} row of A and $\rho(x)$ write an attribute associated A_x by the mapping, ρ. The algorithm randomly chooses $r_{1,x}, r_{2,x}, z_{1,x}, z_{2,x}$ from $\mathbb{Z}_p$ for each $x \in [m]$, and sets $r_x = r_{1,x} + r_{2,x}$ where A is an $m \times \ell$ matrix. It randomly chooses $kTag_{j,x}$ for each $x \in [m]$ and $j \in [n]$ from $\mathbb{Z}_p$. It randomly selects an ℓ size vector $\vec{\mu} \in \mathbb{Z}_p^{\ell}$ of which the first coordinate equals to α, and sets $\lambda_x = A_x\vec{\mu}$ as the share of A_x. It creates SK as follows

$$SK := (D_{1,x}, D_{2,x}, D_{3,x}, D_{4,x}, D_{5,x}, D_{6,x}, D_{7,x},$$

$$\{K_{j,x}, kTag_{j,x}; \forall j \in [n]\}; \forall x \in [m])$$

where

$$D_{1,x} = g_2^{\lambda_x \cdot a_1} v_2^{r_x}, D_{2,x} = g_2^{-\lambda_x} u_2^{r_x} g_2^{z_{1,x}}, D_{3,x} = (g_2^b)^{-z_{1,x}},$$

$$D_{4,x} = {u_2'}^{r_x} g_2^{z_{2,x}}, D_{5,x} = (g_2^b)^{-z_{2,x}}, D_{6,x} = (g_2^b)^{r_{2,x}},$$

$$D_{7,x} = g_2^{r_{1,x}}, K_{j,x} = (g_2^{h_j} g_2^{-h_0\rho(x)^j} w_2^{kTag_{j,x}})^{r_{1,x}}\ \forall j \in [n].$$

Decrypt(SK, CT, PK, $\mathbb{A}$, S) First, the algorithm calculates constants w_x such that $\sum_{\rho(x)\in S} \omega_x A_x = (1, 0, ...0)$. For each $x \in S$, it calculates $c_0, ..., c_n$ by the same manner of Encrypt, then it calculates $Tag_x := \sum_{j=[n]} c_j \cdot kTag_{j,x} - cTag$. If $Tag_x \neq 0$, it calculates

$$W_1 := e(C_1, \prod_{\rho(x)\in S} {D_{1,x}}^{\omega_x}) \cdot e(C_2, \prod_{\rho(x)\in S} {D_{2,x}}^{\omega_x}) \cdots e(C_5, \prod_{\rho(x)\in S} {D_{5,x}}^{\omega_x})$$

$$W_2 := e(C_6, \prod_{\rho(x)\in S} {D_{6,x}}^{\omega}) \cdot e(C_7, \prod_{\rho(x)\in S} {D_{7,x}}^{\omega})$$

$$W_3 := e(E_1, \prod_{\rho(x)\in S} {D_{7,x}}^{-\omega_x/Tag_x}) \cdot e(E_0, \prod_{\substack{\rho(x)\in S,\\ j=[n]}} {K_{j,x}}^{c_j \cdot \omega_x/Tag_x}).$$

Otherwise, it aborts. Finally,

$$M = C \cdot \frac{W_1}{W_2 \cdot W_3}.$$

Correctness can be derived trivially from our scheme with a symmetric pairing.