# Fast, Compact, and Expressive Attribute-Based Encryption

Junichi Tomida
NTT, Japan
junichi.tomida.vw@hco.ntt.co.jp

Yuto Kawahara
NTT, Japan
yuto.kawahara.yk@hco.ntt.co.jp

Ryo Nishimaki
NTT, Japan
ryo.nishimaki.zk@hco.ntt.co.jp

## ABSTRACT

Attribute-based encryption (ABE) is an advanced cryptographic tool and useful to build various types of access control systems. Toward the goal of making ABE more practical, we propose key-policy (KP) and ciphertext-policy (CP) ABE schemes, which first support unbounded sizes of attribute sets and policies with negation and multi-use of attributes, allow fast decryption, and are fully secure under a standard assumption, simultaneously. The proposed schemes are more expressive than previous schemes and efficient enough. We also implement our schemes in 128-bit security level and present their benchmarks for an ordinary personal computer and smartphones. They show that all algorithms run in one second with the personal computer when they handle any policy or attribute set with one hundred attributes.

## KEYWORDS

attribute-based encryption; standard assumption; non-monotone; unbounded; multi-use; random oracle model

## 1 INTRODUCTION

Attribute-based encryption (ABE) [16] is an advanced form of public key encryption (PKE), which yields fine-grained access control over encrypted data. More concretely, ABE allows us to embed an attribute $x$ into a ciphertext when we encrypt a message. An authority that has a master secret key can issue a secret key that is associated with a predicate $y$. The ciphertext can be decrypted with the secret key only if $x$ and $y$ satisfy some relation $R$.

Previously, ABE schemes have been proposed for various relations, such as equality [9], threshold [29], orthogonality of vectors [18], and so on. One of the most notable relations among them is that expressed by an access structure [8, 16]. In a key-policy ABE (KP-ABE) scheme, for instance, one can embed an access structure in a secret key such as (Year:1991-2000 AND Category:jazz). The secret key can decrypt ciphertexts that have attributes Year:1991-2000 and Category:jazz but cannot ones that only have at most one of them. Ciphertext-policy ABE (CP-ABE) is a dual of KP-ABE and allows us to embed an access structure into ciphertexts.

Despite the marvelous functionality of ABE, it does not spread so widely in the real world. Motivated by the situation, Agrawal and Chase recently proposed new KP-ABE and CP-ABE schemes named FAME [1], which are the first schemes that simultaneously:

(1) have no restriction on sizes of policies and attribute sets (unboundedness);
(2) allow an arbitrary string as an attribute (large universe);
(3) are based on the fast Type-III pairings;
(4) need a small number of pairings for decryption;
(5) satisfy the adaptive security under standard assumptions.

All these properties are arguably important in practice. We briefly explain the reasons. The first two properties say about scalability. It is not uncommon that we extend a system to add new attributes to a database in operation. In such cases, scalability is essential property because if the scheme does not have the scalability, we need a redeployment of the scheme. The second two properties say about efficiency. The efficiency of building blocks directly affects that of the entire system. Thus, efficient cryptographic schemes are desirable. The final property says about security. In contrast to the selective security, the adaptive security considers a model that captures a natural attack of an adversary against a scheme. Additionally, standard assumptions are based on well-studied hard problems and thus reliable. Hence, the adaptive security under standard assumptions guarantees that schemes are secure enough.

### 1.1 Our Contribution

Toward the goal to make ABE schemes more usable and realistic, we propose more expressive schemes. More precisely, we propose KP-ABE and CP-ABE schemes that satisfy all the above properties and additionally

(6) allow us to use negation to express access structures *in a natural form* (non-monotonicity);
(7) can handle policies in which the same attributes appear more than once (multi-use of attributes).

These properties allow us to use more fine-grained policies that are commonly used in access control systems in practice. Thanks to great works on ABE [3, 21, 27], we have several ABE schemes that can handle unbounded sizes of attribute sets and policies in prime-order groups. To our knowledge, however, there are no schemes that achieve all the properties listed above simultaneously. We summarize previous schemes and ours in Table 1.

One note is that our schemes require the random oracle model for security analysis as well as FAME. Whereas a random oracle cannot be replaced with any implemented hash function in some particular cases [10], it is still a widely accepted and standard methodology to analyze the security of cryptographic schemes. Actually, many practical schemes that are used in the real world require the random oracle model for their security analysis [6, 7, 14].

In the following, we elaborate on the last two properties.

**Non-monotonicity.** Previously, there are several works that consider access structures including negation (non-monotone access structures) in ABE [3, 4, 24, 26–28, 32]. Among them, only the negation form defined by Okamoto and Takashima (OT negation) [26, 27] is different from that by the others (non-OT negation). Considering an example is the best way to describe the difference. Suppose there are two labels Year and Categoly in KP-ABE, and each attribute is the form like Year:1991-2000. Then, non-OT negation is like (NOT Year:1991-2000) whereas OT negation is like (Year:NOT 1991-2000). Semantically, the former implies that the secret key can decrypt a ciphertext if it does not have attribute

**Table 1: Comparison of unbounded KP and CP-ABE schemes based on prime-order groups.**

| Scheme | Unboundedness | Large universe | Type-III | Fast Dec | Standard assumptions | Non-monotonicity | Multi-use | w/o RO |
|---|---|---|---|---|---|---|---|---|
| OT12 [27] | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | ✓ |
| AC17 [1] | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × |
| KW19 [21] | ✓ | ✓ | ✓ | × | ✓ | × | ✓ | ✓ |
| Att19 [3] | ✓ | ✓ | ✓ | × | × | ×[a] | ✓ | ✓ |
| Ours | ✓ | ✓ | ✓ | ✓[b] | ✓ | ✓ | ✓ | × |

[a] The scheme that is explicitly described by Attrapadung [3] can handle negation, but it is not the natural form that we consider.

[b] The number of pairings in decryption of our schemes does not depend on the size of policies or the number of attributes but only depends on the number of multi-use of labels in a policy. Thus, as long as considering the same setting as FAME, which imposes one-use restriction on policies, the decryption requires only a constant number of pairings.

Year:1991-2000. On the other hand, the latter implies that a ciphertext is decryptable if it has an attribute on label Year and its attribute is not 1991-2000.

When we consider large universe ABE[1], which is exactly the desirable case in practice, the natural negation form is arguably OT negation. In large universe ABE, it is unreasonable to fix all attributes used in a system at the setup phase because the most significant advantage of large universe ABE is that we can utilize an exponentially large number of attributes. Associating strings with attributes that the ABE scheme handles in an ad-hoc way by a hash function would be a better solution. However, if we use non-OT negation in the system, we have to fix all attributes that the system supports at the setup phase. This is because a secret key whose policy is negation of an attribute that the system has not supported before can decrypt all ciphertexts generated so far. More concretely, in the above example, we consider the case where we add a new label Artist in the system. Then, if an authority issues a key whose policy is (NOT Artist:The Beatles), all previous ciphertexts are decrypted by the key even if the underlying content is by The Beatles because they do not have an attribute on label Artist. On the other hand, OT negation does not cause this inconvenience because a key whose policy is (Artist:NOT The Beatles) is useless to decrypt ciphertexts without an attribute on label Artist. Thus, we refer to OT negation as a natural form.

**Multi-use of attributes.** Many ABE schemes whose security relies on the dual system methodology [30] have a one-use restriction on access structures [11, 12, 23, 26, 27]. In an ABE scheme with the one-use restriction, one can use only policies in which all attributes appear once. That is, one cannot embed a policy into a ciphertext or secret key such as ((Year:1991-2000 AND Category:jazz) OR (Year:2001-2010 AND Category:jazz) OR (Year:2001-2010 AND Artist:The Beatles)) because attributes Category:jazz and Year:2001-2010 appear twice in the policy.

One way to circumvent this restriction is to prepare multiple attributes for each attribute in advance like Category:jazz-1, Category:jazz-2, and so on. However, this solution has two problems. The first is that we need to decide the maximum number of identical attributes that appear in a policy at the setup phase. Thus, the access structures that the scheme supports are still limited. The second is that, in KP-ABE, for instance, the solution increases the sizes of ciphertexts proportionally to the maximum number, and it leads to efficiency loss. This fact prevents the solution to set a sufficiently large number for the limit.

On the other hand, in an ABE scheme that supports multi-use of attributes, we have no restrictions on policies and can combine any attributes in an arbitrary way to generate a policy. In KP-ABE, for instance, the sizes of ciphertexts are independent of policies and thus compact.

One caveat is that our schemes only take a policy expressed by a Boolean formula, whereas most ABE schemes support more powerful span programs to express policies. This limitation arises from the security proofs of our schemes. Nevertheless, when we generate a policy, we typically consider it as a Boolean formula and then convert it to a span program [22]. Hence, Boolean formulae seem to be sufficiently expressive in practice.

## 1.2 Design of Our ABE Schemes

In the following, we focus on our KP-ABE scheme. Our relation of ABE is very close to that by Okamoto and Takashima in [27]. An attribute consists of labels, e.g., (Year, Category), and values for each labels, e.g., (1991-2000, jazz). A predicate is an arbitrary Boolean formula that is a combination of variables by operations AND, OR, and NOT such as ((Year:1991-2000 AND Category:jazz) OR (Year:1991-2000 AND Artist:NOT The Beatles)).

Our scheme is based on the dual system encryption, which we can instantiate from either composite-order or prime-order bilinear groups [11, 25, 30, 31]. Our actual scheme is based on prime-order bilinear groups following the framework by Chen et al. [11] to utilize the dual system methodology in prime-order groups and the technique by Agrawal and Chase [1] to utilize a random oracle in asymmetric prime-order bilinear groups. For ease of exposition, we describe the composite-order variant of our scheme in this section. Let $N = p_1 p_2$ for primes $p_1$ and $p_2$, and $(G, H, G_T)$ be bilinear groups of order $N$. Let $g$ and $h$ be generators of $G$ and $H$, and $g_i$ and $h_i$ be generators of subgroups $G_i$ and $H_i$ of order $p_i$ for $i = \{1, 2\}$, respectively. Let $H : \{0,1\}^* \to G_1 \times G_1$ be a hash function modeled as a random oracle, and its input is a label. We denote the output

[1] Actually, all known non-monotone ABE schemes are the large universe construction.

of $H(i)$ by $(g_1^{u_i}, g_1^{h_i})$. Then, our scheme can be written as

$$\mathsf{pk} = (g_1, h_1, e(g_1,h_1)^{\alpha})$$
$$\mathsf{ct} = (h_1^s, \{g_1^{s(x_i u_i + h_i)}\}_{i\in S}, e(g_1,h_1)^{s\alpha} M)$$
$$\mathsf{sk} = \left( \{h_1^{r_i}\}_{i\in[n]}, \left\{ g^{\alpha_i} \cdot g_1^{r_i(y_i u_{\psi(i)} + h_{\psi(i)})} \text{ or } \begin{matrix} g^{-\alpha_i} \cdot g_1^{r_i u_{\psi(i)}}, \\ g^{y_i \alpha_i} \cdot g_1^{r_i h_{\psi(i)}} \end{matrix} \right\}_{i\in[n]} \right),$$

where $S$ is the set of labels, $n$ is the number of variables in the formula, $\psi : [n] \to \{0,1\}^*$ is a function that specifies the label of each variable, $\alpha_i$ is a share of the secret $\alpha$, and $x_i$ and $y_i$ are the values for label $i$. Note that the reason ct and sk contain both elements in $G$ and $H$ is to utilize a hash function in asymmetric groups as FAME [1].

The high-level idea of construction is a combination of secret sharing (SS), identity-based encryption (IBE), and identity-based revocation (IBR). Our scheme can instantiate an arbitrary number of IBE and IBR on the fly by leveraging hash function $H$, and each instance corresponds to each label. A secret key of our scheme consists of secret keys of IBE and IBR, and each secret key hides share $\alpha_i$ of a master secret $\alpha$ generated by SS according to the formula. A ciphertext of ABE consists of ciphertexts of IBE and IBR $\{g_1^{s(x_i u_i + h_i)}\}_{i\in S}$. In decryption, one computes terms $\{e(g_1,h_1)^{s\alpha_i}\}_i$ for labels in which the relation of (in)equality between the ciphertext and secret keys is satisfied. Note that one cannot compute term $e(g_1,h_1)^{s\alpha_i}$ if the relation of (in)equality does not hold in label $i$, thanks to the security of underlying IBE and IBR. If term $e(g_1,h_1)^{s\alpha}$ is recovered via reconstruction of SS, which means that the policy in the secret key is satisfied by the attribute in the ciphertext, one can decrypt the ciphertext of ABE.

By the construction, term $e(g_1,h_1)^{s\alpha_i}$ cannot be computed if a ciphertext of ABE does not contain a ciphertext of IBE and IBE for label $i$. This property serves the natural form of negation described in the previous subsection. Additionally, standard IBE and IBR schemes can issue many secret keys. Thus, one can generate multiple secret keys for the same label, which allow us to handle policies in which the same label appears more than once.

## 1.3 Our Main Technique

We can easily prove the adaptive security of our scheme from a standard assumption by the dual system methodology and the predicate encoding framework as in [31] if $\psi$ is injective, or equivalently the scheme has the one-use restriction. However, if it is not the case, to prove the adaptive security of the scheme from standard assumptions becomes quite difficult and had been a long-standing open problem. Very recently, Kowalczyk and Wee brought a breakthrough for this problem (KW19) [21]. More precisely, they proposed a methodology to prove the adaptive security of the most simple ABE scheme, which supports monotone $\mathsf{NC}_1$ circuits (or equivalently Boolean formulae) for a small attribute universe. The scheme can be written in composite-order groups as

$$\mathsf{pk} = (g_1, h_1, g_1^{w_1}, \ldots, g_1^{w_\ell}, e(g_1,h_1)^{\alpha})$$
$$\mathsf{ct} = (g_1^s, \{g_1^{s w_i}\}_{i\in S}, e(g_1,h_1)^{s\alpha} M)$$
$$\mathsf{sk} = (\{h_1^{r_i}\}_{i\in[n]}, \{h^{\alpha_i} \cdot h_1^{r_i w_{\psi(i)}}\}_{i\in[n]}).$$

Roughly speaking, this scheme can be seen as KP-ABE whose atomic structure is PKE like the ElGamal encryption whereas ours corresponds to IBE and IBR. That is, in the above scheme, whether one can compute term $e(g_1,h_1)^{s\alpha_i}$ in decryption depends on the possession of corresponding secret key $g_1^{s w_i}$.

We briefly recall the framework by KW19. Their framework follows the dual system methodology, which is the standard technique to achieve the adaptive security. In the methodology, we change the challenge ciphertext and secret keys into the semi-functional form. Roughly speaking, semi-functional ciphertexts and secret keys have an additional structure in $G_2$ and $H_2$ as follows:

$$\mathsf{ct} = (g^s, \{g^{s w_i}\}_{i\in S}, e(g,h)^{s\alpha} M)$$
$$\mathsf{sk} = (\{h_1^{r_i}\}_{i\in[n]}, \{h^{\alpha_i} \cdot h_1^{r_i w_{\psi(i)}} \cdot h_2^{\gamma_i}\}_{i\in[n]}),$$

where $\gamma_i$ is a share of a random secret $\gamma$.

In the dual system methodology, we consider a series of hybrids where we first change the challenge ciphertext into the semi-functional form and then the secret keys into the semi-functional form one by one. In the latter part, the methodology allows us to focus on only one secret key by leveraging components in $G_2$ and $H_2$. Therefore, to show the following indistinguishability for the adaptive choice of ct and the one key sk is sufficient to change the target secret key into a semi-functional form:

$$\begin{pmatrix} \mathsf{ct} = (g_2^s, \{g_2^{s w_i}\}_{i\in S}), \\ \mathsf{sk} = (\{h_2^{r_i}\}_{i\in[n]}, \{h_2^{r_i w_{\psi(i)} + \boxed{\gamma_{0,i}}}\}_{i\in[n]}) \end{pmatrix}$$
$$\approx_c \begin{pmatrix} \mathsf{ct} = (g_2^s, \{g_2^{s w_i}\}_{i\in S}), \\ \mathsf{sk} = (\{h_2^{r_i}\}_{i\in[n]}, \{h_2^{r_i w_{\psi(i)} + \boxed{\gamma_{1,i}}}\}_{i\in[n]}) \end{pmatrix},$$

where $\gamma_{0,i}$ is a share of secret 0 and $\gamma_{1,i}$ is a share of secret $\gamma$. This core component is called core 1-ABE.

The difficulty of showing the indistinguishability of core 1-ABE from a standard assumption arises from the fact that we need to embed a computational problem into sk depending on ct. That is, if an adversary first asks for sk, a simulator has no idea on how to embed the computational problem into sk. Their framework tells us how to construct a series of hybrids to show the above indistinguishability. In each transition of hybrids, the simulator guesses a part of the adversary's choice that has sufficient information to embed the problem into sk. Simultaneously, the part must be so small that the simulator can guess it with a non-negligible probability. In our case, the part tells the correct element in sk where the simulator embeds the problem. Observe that each $\gamma_i$ is hidden by a kind of ElGamal encryption in $H_2$. Thus, we can embed the DDH problem based on the guess and gradually change shares $\{\gamma_i\}_{i\in[n]}$.

At a glance, their framework seems applicable to our scheme directly, but actually, it does not work. The main problem is the fact that whereas their framework tells us the location and its label where we should embed the problem in sk, it does not tell us the value of the label in ct. In other words, the difficulty of directly applying their framework to our scheme seems essentially the same as that of proving the adaptive security of Boneh-Boyen IBE, which was proven secure only in the selective setting. This problem does not occur in the scheme by KW19 because the corresponding part is just a kind of ElGamal-like encryption.

To overcome the problem, we introduce new usage of the framework by KW19 that allows us to utilize the dual system methodology more beneficially. As we mentioned previously, a secret key of our scheme contains many secret keys of IBE and IBR based on the dual system encryption. Furthermore, the framework tells us which secret key should be changed in each hybrid in the core 1-ABE. Thus, we can gradually randomize the component in $H_2$ of each element in sk by the dual system methodology instead of the DDH problem in $H_2$.

For simplicity, we show the case where we apply our new technique to the scheme by KW19. In our technique, we consider the following indistinguishability of core 1-ABE:

$$\begin{pmatrix} \mathsf{ct} = (g^s, \{g^{s w_i}\}_{i\in S}), \\ \\ \mathsf{sk} = (\{h_1^{r_i}\}_{i\in[n]}, \{h_1^{r_i w_{\psi(i)}} \cdot h_2^{\boxed{\gamma_{0,i}}}\}_{i\in[n]}) \end{pmatrix} \approx_c \begin{pmatrix} \mathsf{ct} = (g^s, \{g^{s w_i}\}_{i\in S}), \\ \\ \mathsf{sk} = (\{h_1^{r_i}\}_{i\in[n]}, \{h_1^{r_i w_{\psi(i)}} \cdot h_2^{\boxed{\gamma_{1,i}}}\}_{i\in[n]}) \end{pmatrix}.$$

We use the dual system methodology to randomize the component in $H_2$. Let $i^*$ be the location where $\gamma_{i^*}$ is supposed to be changed in some two hybrids, which means that $i^* \notin S$. Then, from the sub-group assumption, the dual system methodology argue that

$$(h_1^{r_{i^*}}, h_1^{r_{i^*} w_{\psi(i^*)}} \cdot h_2^{\gamma_{i^*}}) \approx_c (h^{r_{i^*}}, h^{r_{i^*} w_{\psi(i^*)}} \cdot h_2^{\gamma_{i^*}}).$$

Then, we can observe that $w_{\psi(i^*)} \mod p_2$ in sk is randomly distributed in $\mathbb{Z}_{p_2}$ from the Chinese remainder theorem and the fact $i^* \notin S$. Thus, term $\gamma_i$ is completely hidden by term $r_{i^*} w_{\psi(i^*)}$. Unlike the framework by KW19, we can apply this technique to our scheme similarly. This is because the simulator does not need to know the value of label $i^*$ in ct as in the proof of IBE based on the dual system encryption.

## 1.4 Other Techniques

Furthermore, we give the following technical contributions:

- reducing the number of pairings in decryption;
- reducing the number of shares of secret sharing;
- making the proof simpler;
- presenting our CP-ABE scheme.

**Number of pairings.** Our scheme described in Section 1.2 requires $O(n)$ pairings in decryption. To reduce the number, we employ the construction by Agrawal and Chase in [2]. That is, we use an exponent $r_{\pi(i)}$ instead of $r_i$, where $\pi(i) = |\{j \mid \psi(j) = \psi(i), j \le i\}|$. In this construction, we need $O(d)$ pairings in decryption where $d = \max \pi(i)$. In other words, the number of pairings only depends on the maximum number of multi-use of labels in the policy of the secret key. Because our scheme in prime-order groups follows the construction, it allows fast decryption for secret keys with a small number of multi-use of labels.

**Number of shares.** In the scheme by KW19, they use a secret sharing scheme where the number of shares corresponds to the summation of the numbers of gates and input wires when we capture a Boolean formula as a circuit. On the other hand, our schemes employ a secret sharing scheme where the number of shares corresponds to only the number of input wires. Their framework derives from the technique to prove the adaptive security of secret sharing for monotone circuits by Jafargholi et al. [17], which requires the same number of shares as in KW19. We guess that this is why their construction employs such a secret sharing scheme. However, we show that we do not need shares for the gates in secret sharing schemes for Boolean formulae to utilize the framework.

**Simpler proof.** Our scheme follows the technique of FAME to make our scheme unbounded by a hash function [1]. In their construction, they add a term $\sigma \mathbf{a}^\perp$ for randomly chosen $\sigma$ to each element in a secret key because the term is necessary for the security proof. We show that we can generate the corresponding term by a pseudorandom function (PRF), and this construction significantly ease the security proof. Concretely, we can skip the part that corresponds to $\mathsf{Hyb}_0$ to $\mathsf{Hyb}_{2,3,q}$ in their security proof [1, Appendix C]. Note that the additional computational cost by the modification is quite small compared with the whole procedure of the key generation because it requires only small numbers of PRF evaluations and multiplications in $\mathbb{Z}_p$ for each element in a secret key.

**CP-ABE scheme.** We present our CP-ABE scheme and its security proof. Note that the security proof of our CP-ABE scheme is more complicated than that of our KP-ABE scheme, because we need two hidden spaces as in [12, 15] due to a technical reason.

## 1.5 Implementation and Evaluation

We implement our KP-ABE and CP-ABE schemes in 128-bit security level and measure benchmarks for an ordinary personal computer and two smartphones: iOS-based and Android-based. In our schemes, a running time of each algorithm is affected by the numbers of negation and multi-use of labels in a policy as well as the number of attributes. To show the effects of these factors, we present benchmarks for four types of policies that differ in the existence of negation and multi-use.

We roughly describe the running times of our schemes when we handle a policy or attribute set with 100 attributes on a personal computer. In all cases, our KP-ABE (resp. CP-ABE) scheme takes about 0.4 to 0.7s (resp. 0.4 to 0.9s) for encryption and key generation. Decryption is heavily affected by a type of policy, and our schemes take only about 0.02s (KP & CP) in the fastest case and 0.5 (KP) or 0.7s (CP) even in the slowest case. Thus, we can conclude that our schemes take less than 1s in any process and any cases with 100 attributes.

# 2 PRELIMINARY

## 2.1 Notation

For a natural number $n \in \mathbb{N}$, $[n]$ denotes a set $\{1, \ldots, n\}$. For a set $S$, $s \leftarrow S$ denotes that $s$ is uniformly chosen from $S$. For matrices with the same number of rows $\mathbf{A}_1$ and $\mathbf{A}_2$, $(\mathbf{A}_1||\mathbf{A}_2)$ denotes the matrix generated by their concatenation. We denote the whole space spanned by all columns of matrix $\mathbf{A}$ by $\mathsf{span}(\mathbf{A})$. For a matrix $\mathbf{A} := (a_{j,\ell})_{j,\ell}$ over $\mathbb{Z}_p$, $[\mathbf{A}]_i$ $(i \in \{1, 2, T\})$ denotes a matrix over $G_i$ whose $(j, \ell)$ entry is $g_i^{a_{j,\ell}}$, and we apply the similar notation to

vectors and scalars. We denote $([\mathbf{A}]_1, [\mathbf{A}]_2)$ by $[\mathbf{A}]_{1,2}$. For matrices $\mathbf{A}$ and $\mathbf{B}$ where $\mathbf{A}^\top\mathbf{B}$ is defined, we abuse the pairing notation in the following way: $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{A}^\top\mathbf{B}]_T$. A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and denotes $f(\lambda) \leq \mathsf{negl}(\lambda)$. For families of distributions $X := \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y := \{Y_\lambda\}_{\lambda\in\mathbb{N}}$, $X \approx_c Y$ means that they are computationally indistinguishable.

## 2.2 Basic Tools

*Definition 2.1 (Boolean Formula).* A Boolean formula is combination of Boolean variables by binary operations AND and OR, and unary operation NOT. It is well known that a Boolean formula can be represented by a Boolean circuit whose all gates have fan-in 2 and fan-out 1. Throughout the paper, we treat Boolean formulae as the latter form. We refer to an input wire of a Boolean formula as input wire. On the other hand, we refer to an input wire of each gate as incoming wire. Similarly, we refer to the output wire of a formula as output wire and the output wire of each gate as outgoing wire. Additionally, we refer to a gate with input wires as input gate and the gate with the output wire as output gate. A monotone Boolean formula consists of only AND and OR gates, whereas a non-monotone Boolean formula additionally contains NOT gates. Standard complexity theory tells us that any Boolean formula can be converted into equivalent one with a logarithmic depth.

*Definition 2.2 (Pseudorandom Functions).* A pseudorandom function (PRF) family $\mathcal{F} := \{F_K\}_{K\in\mathcal{K}_\lambda}$ with a key space $\mathcal{K}_\lambda$, a domain $\mathcal{X}_\lambda$, and a range $\mathcal{Y}_\lambda$ is a function family that consists of functions $F_K : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. Let $\mathcal{R}_\lambda$ be a set of functions consisting of all functions whose domain and range are $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ respectively. For any PPT adversary $\mathcal{A}$, the following condition holds,

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}^{F_K(\cdot)}] - \Pr[1 \leftarrow \mathcal{A}^{R(\cdot)}]| \leq \mathsf{negl}(\lambda),$$

where $K \leftarrow \mathcal{K}_\lambda$ and $R \leftarrow \mathcal{R}_\lambda$.

*Definition 2.3 (Bilinear Groups).* A description of bilinear groups $\mathbb{G}:=(p, G_1, G_2, G_T, g_1, g_2, e)$ consist of a prime $p$, cyclic groups $G_1$, $G_2, G_T$ of order $p$, generators $g_1$ and $g_2$ of $G_1$ and $G_2$ respectively, and a bilinear map $e : G_1 \times G_2 \to G_T$, which has two properties.

- (Bilinearity): $\forall h_1 \in G_1, h_2 \in G_2, a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$.
- (Non-degeneracy): For generators $g_1$ and $g_2$, $g_T := e(g_1, g_2)$ is a generator of $G_T$.

A bilinear group generator $\mathcal{G}_{\mathsf{BG}}(1^\lambda)$ takes a security parameter $1^\lambda$ and outputs a description of bilinear groups $\mathbb{G}$ with a $\mathsf{poly}(\lambda)$ bit prime. In this paper, we refer to Type-I groups, where efficient isomorphisms exist in both way between $G_1$ and $G_2$, as symmetric bilinear groups, and Type-III groups, where no efficient isomorphisms exist between them, as asymmetric bilinear groups.

For the proofs of our schemes, we utilize the $\mathcal{D}_k$-MDDH assumption [13], which is generalization of the DDH assumption. There are mainly two types of $\mathcal{D}_k$-MDDH assumption families for asymmetric bilinear groups. In the first one, an instance contains unilateral group elements such as the SXDH assumption. The other one consists of assumptions that are involved with bilateral group elements such as the DLIN assumption used in [1], which is sometimes called the XDLIN assumption. In our paper, we utilize the latter type.

*Definition 2.4 ($\mathcal{D}_{j,k}$-MDDH Assumption).* For $j > k$, let $\mathcal{D}_{j,k}$ be a matrix distribution over full rank matrices in $\mathbb{Z}_p^{j\times k}$. We can assume that, wlog, the first $k$ rows of a matrix $\mathbf{A}$ chosen from $\mathcal{D}_{j,k}$ form an invertible matrix. We consider the following distribution:

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda),\ \ \mathbf{A} \leftarrow \mathcal{D}_k,\ \ \mathbf{v} \leftarrow \mathbb{Z}_p^k,\ \ \mathbf{t}_0 := \mathbf{A}\mathbf{v},\ \ \mathbf{t}_1 \leftarrow \mathbb{Z}_p^j,$$
$$P_\beta := (\mathbb{G}, [\mathbf{A}]_{1,2}, [\mathbf{t}_\beta]_{1,2}).$$

We say that the bilateral $\mathcal{D}_{j,k}$-MDDH assumption holds with respect to $\mathcal{G}_{\mathsf{BG}}$ if, for any PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathcal{D}_{j,k}\text{-MDDH}}_{\mathcal{A},\mathsf{bi}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(P_0)] - \Pr[1 \leftarrow \mathcal{A}(P_1)]| \leq \mathsf{negl}(\lambda).$$

We denote $\mathcal{D}_{k+1,k}$ by $\mathcal{D}_k$. Let $\mathcal{U}_{j,k}$ be a uniform distribution over full rank matrices in $\mathbb{Z}_p^{j\times k}$. Then, the following relations hold with tight reductions; $\mathcal{D}_k$-MDDH $\Rightarrow$ $\mathcal{U}_k$-MDDH $\Rightarrow$ $\mathcal{U}_{j,k}$-MDDH.

For an appropriate distribution $\mathcal{D}_k$, the $\mathcal{D}_k$-MDDH assumption generically holds in $k$-linear groups [13]. Thus, in asymmetric bilinear groups, we can utilize the bilateral $\mathcal{D}_k$-MDDH assumption for $k \geq 2$.

*Definition 2.5 (Matrix Notation).* For a matrix $\mathbf{A} \in \mathcal{D}_k$, we define a matrix $\mathbf{A}^*$, and vectors $\mathbf{a}_\mathsf{R}$ and $\mathbf{a}^\perp$ as follows. The vector $\mathbf{a}_\mathsf{R}$ is a vector deterministically computed from $\mathbf{A}$ in a fixed manner so that $\overline{\mathbf{A}} := (\mathbf{A}||\mathbf{a}_\mathsf{R})$ forms a basis. $\mathbf{A}^*$ and $\mathbf{a}^\perp$ are the matrix that consists of the left $k$ columns of $(\overline{\mathbf{A}}^\top)^{-1}$ and the vector that consists of right one column of $(\overline{\mathbf{A}}^\top)^{-1}$, respectively. Note that the following relations hold, namely, $\mathbf{A}^\top\mathbf{A}^* = \mathbf{I}_k$, $\mathbf{A}^\top\mathbf{a}^\perp = \mathbf{0}$, and $\mathbf{A}^*\mathbf{A}^\top + \mathbf{a}^\perp\mathbf{a}_\mathsf{R}^\top = \mathbf{I}_{k+1}$.

## 2.3 Attribute-Based Encryption

*Definition 2.6 (Attribute-Based Encryption).* An attribute-based encryption (ABE) scheme for relation $R : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ consists of four algorithms, where $\mathcal{X}$ and $\mathcal{Y}$ are an attribute universe and predicate universe, respectively.

$\mathsf{Setup}(1^\lambda)$**:** It takes a security parameter $1^\lambda$ and outputs a public key pk and a master secret key msk. pk specifies a message space $\mathcal{M}$.

$\mathsf{Enc}(\mathsf{pk}, x, m)$**:** It takes pk, an attribute $x \in \mathcal{X}$ and a message $m \in \mathcal{M}$ and outputs a ciphertext $\mathsf{ct}_x$.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$**:** It takes pk, msk, and a predicate $y \in \mathcal{Y}$ and outputs a secret key $\mathsf{sk}_y$.

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)$**:** It takes pk, $\mathsf{ct}_x$ and $\mathsf{sk}_y$ and outputs a message $m'$ or a symbol $\perp$.

**Correctness.** An ABE scheme is *correct* if it satisfies the following condition. For all $\lambda \in \mathbb{N}$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$ such that $R(x, y) = 1$, and $m \in \mathcal{M}$, we have

$$\Pr\left[m = m' \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)\\ \mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{pk}, x, m)\\ \mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)\\ m' := \mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)\end{array}\right] = 1.$$

**Security.** An ABE scheme is *adaptively secure* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ defined as follows

is negligible in $\lambda$ for all stateful PPT adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda) := \left| \Pr\left[ \beta = \beta' \;\middle|\; \begin{array}{l} \beta \leftarrow \{0,1\} \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (x^*, m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot)}(\mathsf{pk}) \\ \mathsf{ct}_{x^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, x^*, m_\beta) \\ \beta' \leftarrow \mathcal{A}(\mathsf{ct}_{x^*}) \end{array} \right] - \frac{1}{2} \right|,$$

where all predicates $\{y_i\}_{i \in [q_{\mathsf{sk}}]}$ on which $\mathcal{A}$ queries KeyGen must satisfy $R(x^*, y_i) = 0$.

A relations for ABE that we consider in our paper is expressed by a non-monotone Boolean formula over the equivalence relation in $\mathbb{Z}_p$. More specifically, each input of the Boolean formula is decided by whether a certain components in a attribute and predicate are equal. Then, the relation is decided by the output of the formula. Our relation is very close to that formulated by Okamoto and Takashima in [27], though their scheme has one-use restriction on labels in policies. One caveat is that we can use only a non-monotone Boolean formula for a predicate in our scheme, whereas the relation by Okamoto and Takashima allows us to use a more powerful non-monotone span program for a predicate. In the following, we consider only non-monotone Boolean formulae where NOT gates exist only on input wires, and any non-monotone formula can be easily converted into such a formula.

*Definition 2.7 (Relation R).* Relation $R$ for our schemes is defined as follows.

- $\mathcal{X} = \bigcup_{i \in \mathbb{N}} \mathbb{Z}_p^i \times \Phi_i$, where $\Phi_i$ consists of all injective functions such that $\phi : [i] \to \{0,1\}^*$.
- $\mathcal{Y} = \bigcup_{i \in \mathbb{N}} \mathbb{Z}_p^i \times \mathcal{F}_i \times \Psi_i \times \mathcal{T}_i$, where $\mathcal{F}_i$ consists of all monotone Boolean formulae whose input lengths are $i$, and $\Psi_i$ and $\mathcal{T}_i$ consist of all functions such that $\psi : [i] \to \{0,1\}^*$ and $t : [i] \to \{0,1\}$, respectively.
- For $x = (\mathbf{x} \in \mathbb{Z}_p^m, \phi)$ and $y = (\mathbf{y} \in \mathbb{Z}_p^n, f, \psi, t)$, we define $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ as

$$b_i := \begin{cases} t(i) \odot \mathsf{true}(x_{\phi^{-1}(\psi(i))} = y_i) & \psi(i) \subseteq \mathrm{Im}(\phi) \\ 0 & \psi(i) \nsubseteq \mathrm{Im}(\phi) \end{cases},$$

  where $\odot$ denotes xnor. Then, $R(x, y) = 1 \Leftrightarrow f(b) = 1$.

The above definition is the relation for key-policy ABE (KP-ABE), and in ciphertext-policy ABE (CP-ABE), $\mathcal{X}$ and $\mathcal{Y}$ are opposite.

For $\mathcal{X}$, each element of $\mathbf{x} \in \mathbb{Z}_p^m$ corresponds to a value for some label, and $\phi$ specifies which label each element of $\mathbf{x}$ is associated with. For instance, when we consider an attribute (Age: 22, Hobby: tennis), $x = (\mathbf{x}, \phi)$ can be set as $\mathbf{x} := (22, H_1(\text{tennis}))$, $\phi(1) :=$ Age, and $\phi(2) :=$ Hobby with a collision resistant hash function $H_1 : \{0,1\}^* \to \mathbb{Z}_p$.

For $\mathcal{Y}$, each element of $\mathbf{y} \in \mathbb{Z}_p^n$ corresponds to the value for each input wire of $f$, and $\psi$ specifies which label each input wire of $f$ is associated with. Additionally, $t$ specifies whether each input wire corresponds to affirmation or negation. For instance, let us consider a predicate (Age = 25 AND Hobby ≠ baseball). Then, $y = (\mathbf{y}, f, \psi, t)$ can be set as $\mathbf{y} := (25, H_1(\text{baseball}))$, $f$ is an AND gate, $\psi(1) :=$ Age and $\psi(2) :=$ Hobby, and $t(1) = 1$ and $t(2) = 0$.

*Definition 2.8 (Linear Secret Sharing Scheme).* A linear secret sharing scheme (LSSS) for a function class $\mathcal{F}$ consists of two algorithms Share and Rec.

**Share$(f, \mathbf{k})$:** It takes a function $f \in \mathcal{F}$ where $f : \{0,1\}^n \to \{0,1\}$ and a vector $\mathbf{k} \in \mathbb{Z}_p^\ell$. Then, outputs shares $\mathbf{k}_1, \ldots, \mathbf{k}_n \in \mathbb{Z}_p^\ell$.

**Rec$(f, x, \{\mathbf{k}_i\}_{x_i=1})$:** It takes $f : \{0,1\}^n \to \{0,1\}$, a bit string $x := (x_1, \ldots, x_n) \in \{0,1\}^n$ and shares $\{\mathbf{k}_i\}_{x_i=1}$. Then, outputs a vector $\mathbf{k}'$ or $\perp$.

In particular, Rec computes a linear function on shares to reconstruct a secret; $\mathbf{k} = \sum_{x_i=1} a_i \mathbf{k}_i$ where each $a_i$ is determined by $f$. A LSSS has two properties.

**Correctness:** For any $f \in F$, $x \in \{0,1\}^n$ such that $f(x) = 1$,

$$\Pr[\mathsf{Rec}(f, x, \{\mathbf{k}_i\}_{x_i=1}) = \mathbf{k} \mid \mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k})] = 1.$$

**Security:** For any $f \in F$, $x \in \{0,1\}^n$ such that $f(x) = 0$, and $\mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k})$, shares $\{\mathbf{k}_i\}_{x_i=1}$ have no information about $\mathbf{k}$.

## 2.4 Piecewise Guessing Framework

Here, we briefly recall the piecewise guessing framework by Kowalczyk and Wee [21], which is based on the framework by Jafargholi et al. [17]. The framework helps us to prove adaptive security of cryptographic schemes that are selectively secure.

*Definition 2.9 (Interactive Game).* An interactive game G is a game between an adversary $\mathcal{A}$ and a challenger $C$. In the game, $\mathcal{A}$ and $C$ send messages interactively, and the messages sent by $C$ depend on the game G. After the interaction, $\mathcal{A}$ outputs $\beta \in \{0,1\}$. We denotes the output of $\mathcal{A}$ in G by $\langle \mathcal{A}, \mathrm{G} \rangle$. Let $z \in \{0,1\}^R$ be a part of messages supposed to be sent by $\mathcal{A}$ in the game. In the adaptive game G, $\mathcal{A}$ can send $z$ at arbitrary points as long as it follows a rule of the game. We define the selective variant of G, denoted by $\widehat{\mathrm{G}}$, to be the same as G except that $\mathcal{A}$ has to declare $z$ that will be sent in the game, at the beginning of the interaction.

Suppose we want to show that adaptive games $\mathrm{G}_0$ and $\mathrm{G}_1$ are computationally indistinguishable, i.e.,

$$|\Pr[\langle \mathcal{A}, \mathrm{G}_0 \rangle = 1] - \Pr[\langle \mathcal{A}, \mathrm{G}_1 \rangle = 1]| \leq \mathsf{negl}(\lambda).$$

Then, we consider a series of selective hybrids $\widehat{\mathrm{H}}^{h_0}, \ldots, \widehat{\mathrm{H}}^{h_L}$ such that

$$\widehat{\mathrm{G}}_0 = \widehat{\mathrm{H}}^{h_0} \approx_c \widehat{\mathrm{H}}^{h_1} \approx_c, \ldots, \approx_c \widehat{\mathrm{H}}^{h_L} = \widehat{\mathrm{G}}_1,$$

where $h_0, \ldots, h_L : \{0,1\}^R \to \{0,1\}^{R'}$ for some $R' \ll R$, and $\widehat{\mathrm{H}}^{h_\iota}$ is an interactive game in which $C$'s messages depend on $u := h_\iota(z)$. Additionally, $h_0$ and $h_L$ need to be constant functions. Note that $C$ can generate messages depending on $u$ because $z$ is declared at the beginning of the interaction. Next, we define variants of $\widehat{\mathrm{H}}^{h_\iota}$, namely, $\widehat{\mathrm{H}}^{h_\iota}_0$ and $\widehat{\mathrm{H}}^{h_\iota}_1$ as follows. In $\widehat{\mathrm{H}}^{h_\iota}_\beta$ for $\beta \in \{0,1\}$, $\mathcal{A}$ has to declare $h_{\iota-1+\beta}(z)$ and $h_{\iota+\beta}(z)$ instead of $z$ at the beginning of the game. Then, $C$ interacts with $\mathcal{A}$ setting $u := h_\iota(z)$ in both $\widehat{\mathrm{H}}^{h_\iota}_0$ and $\widehat{\mathrm{H}}^{h_\iota}_1$. In other words, $\widehat{\mathrm{H}}^{h_\iota}_\beta$ is the same as $\widehat{\mathrm{H}}^{h_\iota}$ except that only partial information of $z$ is declared. Now we are ready to state the adaptive security lemma.

Lemma 2.10 (Adaptive Security Lemma [21]). *Let $\mathrm{G}_0$ and $\mathrm{G}_1$ be adaptive interactive games and $\{\widehat{\mathrm{H}}^{h_i}\}_{0 \leq i \leq L}$ be selective hybrids defined above. Suppose they satisfy the two properties:*

- $\mathrm{G}_0 = \mathrm{H}^{h_0}$ *and* $\mathrm{G}_1 = \mathrm{H}^{h_L}$*, where* $\mathrm{H}^{h_0}$ *and* $\mathrm{H}^{h_L}$ *are the same as* $\widehat{\mathrm{H}}^{h_0}$ *and* $\widehat{\mathrm{H}}^{h_L}$*, respectively, except that* $\mathcal{A}$ *does not declare* $z$ *at the beginning. Note that* $C$*'s messages can be correctly defined because* $h_0$ *and* $h_L$ *are constant functions.*
- *For all PPT adversary* $\mathcal{A}$ *and all* $\iota \in L$*, we have*

$$|\Pr[\langle \mathcal{A}, \widehat{\mathrm{H}}_1^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathrm{H}}_0^{h_\iota} \rangle = 1]| \leq \epsilon.$$

*Then, we have*

$$|\Pr[\langle \mathcal{A}, \mathrm{G}_0 \rangle = 1] - \Pr[\langle \mathcal{A}, \mathrm{G}_1 \rangle = 1]| \leq 2^{2R'} L\epsilon.$$

## 2.5 Pebbling Strategy for Boolean Formula

A pebbling strategy is used for a guide of how to construct a series of hybrids in the piecewise guessing framework.

*Definition 2.11 (Pebbling Game).* A player of the pebbling game is given a monotone Boolean formula $f : \{0,1\}^n \to \{0,1\}$ and input $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ such that $f(b) = 0$. The goal of the game is to reach the state where a pebble is placed on only the output gate, starting from the state with no pebbles on the Boolean formula $f$, following a pebbling rule. The rule is defined as follows.

(1) We can place or remove a pebble on an AND gate if at least one of its incoming wires comes from a gate or input wire with a pebble on it.
(2) We can place or remove a pebble on an OR gate if both of its incoming wires come from a gate or input wire with a pebble on it, respectively.
(3) We can place or remove a pebble on input wire $i$ whose input corresponds to 0, i.e., $b_i = 0$.
(4) We can pass the turn, which allows us to increase the total number of steps in the game without changing the pebbling strategy.

*Definition 2.12 (Pebbling Record).* A pebbling record $\mathcal{R} := (r_0, \ldots, r_L) \in (\{0,1\}^{R'})^L$ is a list of all pebbling configuration that a player took from the start to the goal in the game. $R'$-bit string $r_\iota$ specifies the configuration at the $\iota$-th step in the play. Thus, $r_0$ specifies the state with no pebbles and $r_L$ specifies the state with one pebble on the output gate. It also means that the player takes $L$ steps to reach the goal. Furthermore, all pebbling configurations that the player took can be specified by an $R'$-bit string.

The following lemma says that, for any monotone Boolean formula and input, there exists a pebbling strategy where all pebbling configurations can be specified with a "short" bit string.

Lemma 2.13 (Pebbling Lemma[21]). *Let* $f : \{0,1\}^n \to \{0,1\}$ *be any monotone Boolean formula with a depth* $d \leq B$*, and* $b \in \{0,1\}^n$ *be any bit string such that* $f(b) = 0$*. Then, there exists a deterministic algorithm* $\mathsf{PebRec}(f, b)$ *that takes* $f$ *and* $b$ *and outputs a record* $\mathcal{R}$ *consisting of* $8^B$ *strings whose lengths are* $3B$ *bits.*

# 3 ADAPTIVE SECURITY FOR CORE COMPONENT

In this section, we present a main lemma for the security proofs of our ABE schemes. We use the piecewise guessing framework and algorithm PebRec for the proof of the main lemma. The proofs of ABE schemes basically follow the dual system methodology, in which we employ a series of hybrids. In these hybrids, we first change the challenge ciphertext into a so-called semi-functional form. Then, we also change all secret keys into the semi-functional form one by one. The main lemma is used to show the indistinguishability of hybrids in the latter part.

Let us focus on our KP-ABE scheme. Roughly speaking, the main lemma implies the indistinguishability between the normal and semi-functional secret keys in the game where an adversary can obtain a core component adaptively, which consists of an ingredient of ciphertext and one of the two secret keys. The core component is called core 1-ABE.

Before we move to the lemma, we describe a linear secret sharing scheme that we use in our scheme as a building block.

## 3.1 Linear Secret Sharing for Boolean Formulae

Our secret sharing scheme for Boolean formulae is described in Fig 1, which is essentially the same as the scheme in [22, Appendix G]. Observe that it works similarly if all vectors in Fig 1 are group elements. Let $f$ be a formula and $b = (b_1, \ldots, b_n)$ be a bit string such that $f(b) = 1$. Then, for reconstruction, it is not difficult to see that there exists a set $S \subseteq \{i \mid b_i = 1\}$ such that $\sum_{i \in S} \boldsymbol{\sigma}_i = \mathbf{k}$.

Clearly, the number of shares for formula $f$ corresponds to the number of its input wires. The secret sharing scheme employed by Kowalczyk and Wee is different from ours [21], where the number of shares corresponds to the summation of the numbers of input wires and gates in $f$. We show that we can utilize the piecewise guessing framework to show the indistinguishability of core 1-ABE even if we replace the scheme to ours.

$\underline{\mathsf{Share}(f, \mathbf{k})}$

Input: A formula $f : \{0,1\}^n \to \{0,1\}$ and a secret $\mathbf{k} \in \mathbb{Z}_p^\ell$.

(1) Set a vector $\boldsymbol{\sigma}_{\mathsf{out}} := \mathbf{k}$ on the output wire.
(2) For each AND gate $g$ with incoming wires $a, b$ and an outgoing wire $c$ where a vector $\boldsymbol{\sigma}_c$ is set on $c$, choose $\mathbf{u}_g \leftarrow \mathbb{Z}_p^\ell$ and set $\boldsymbol{\sigma}_a := \boldsymbol{\sigma}_c - \mathbf{u}_g$ and $\boldsymbol{\sigma}_b := \mathbf{u}_g$ on $a$ and $b$, respectively.
(3) For each OR gate $g$ with incoming wires $a, b$ and an outgoing wire $c$ where a vector $\boldsymbol{\sigma}_c$ is set on $c$, set $\boldsymbol{\sigma}_a := \boldsymbol{\sigma}_c$ and $\boldsymbol{\sigma}_b := \boldsymbol{\sigma}_c$ on $a$ and $b$, respectively.
(4) Output shares $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n$, which are set on the input wires $1, \ldots, n$.

**Figure 1: Our linear secret sharing scheme for Boolean formulae.**

We have the following lemma for the secret sharing scheme.

Lemma 3.1. *For all* $\ell, n \in \mathbb{N}$*, Boolean formulae* $f : \{0,1\}^n \to \{0,1\}$*,* $\mathbf{k}, \mathbf{a} \in \mathbb{Z}_p^\ell$*, and* $\mu \in \mathbb{Z}_p$*, we define the following distribution.*

$$\mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k} + \mu\mathbf{a}), \quad \mathbf{k}'_1, \ldots, \mathbf{k}'_n \leftarrow \mathsf{Share}(f, \mathbf{k}),$$
$$\sigma_1, \ldots, \sigma_n \leftarrow \mathsf{Share}(f, \mu).$$

*Then, the two distributions are identical:*

$$\{\mathbf{k}_1, \ldots, \mathbf{k}_n\} \text{ and } \{\mathbf{k}'_1 + \sigma_1\mathbf{a}, \ldots, \mathbf{k}'_n + \sigma_n\mathbf{a}\}.$$

Proof. Let $g^*$ be a gate in $f$ with incoming wires $a, b$ and outgoing wire $c$. Let $\boldsymbol{\sigma}_i$ for $i \in \{a, b, c\}$ be values set on a wire $i$ in

the execution of $\mathsf{Share}(f, \mathbf{x})$. From the procedure of the scheme, we have $\boldsymbol{\sigma}_i = b_{\mathsf{out}}\mathbf{x} + \sum_{g\in S} b_g \mathbf{u}_g$ for some subset $S$ of all gates in $f$ and $b_{\mathsf{out}}, b_g \in \{0,1\}$. Note that $S$, $b_{\mathsf{out}}$, $b_g$ are determined by $f$ and $i$.

Let $\mathbf{k}_i$, $\mathbf{k}'_i$, and $\sigma_i$ for $i \in \{a, b, c\}$ be values set on wire $i$ in the execution of $\mathsf{Share}(f, \mathbf{k}+\mu\mathbf{a})$, $\mathsf{Share}(f, \mathbf{k})$, and $\mathsf{Share}(f, \mu)$, respectively. Then, we have

$$\mathbf{k}_i = b_{\mathsf{out}}(\mathbf{k}+\mu\mathbf{a}) + \sum_{g\in S} b_g \mathbf{u}_g,$$
$$\mathbf{k}'_i = b_{\mathsf{out}}\mathbf{k} + \sum_{g\in S} b_g \mathbf{u}'_g,$$
$$\sigma_i = b_{\mathsf{out}}\mu + \sum_{g\in S} b_g u_g,$$

for some randomly chosen $\mathbf{u}_g$, $\mathbf{u}'_g$, and $u_g$. We can implicitly define $\mathbf{u}_g := \mathbf{u}'_g + u_g\mathbf{a}$, and thus $\mathbf{k}_i = \mathbf{k}'_i + \sigma_i\mathbf{a}$ for $i \in \{\text{all wires in } f\}$. This concludes the proof. □

## 3.2 Core Component

*Definition 3.2 (Core 1-ABE).* We define $\mathsf{G}^{\text{1-ABE}}_\beta$ for $\beta \in \{0,1\}$ as Fig 2. In $\mathsf{G}^{\text{1-ABE}}_\beta$, $\mathcal{A}$ can make a query to $O_X$ and $O_F$ only once whereas $\mathcal{A}$ can do to $O_R$ polynomial times. All queries can be done adaptively. Furthermore, $x$ and $y$ on which $\mathcal{A}$ queries $O_X$ and $O_F$ must satisfy $R(x,y) = 0$. $\mathcal{X}$ and $\mathcal{Y}$ are defined in Definition 2.7. Note that the difference between $\mathsf{G}^{\text{1-ABE}}_0$ and $\mathsf{G}^{\text{1-ABE}}_1$ lies in the input of Share in $O_F$.

Note that $O_X$ outputs the ingredient of the challenge ciphertext, $O_F$ outputs the target secret key, and $O_R$ is used for simulating a random oracle.

Lemma 3.3 (Core 1-ABE Security). *Let $B$ be the maximum depth of formula $f$ for all choice of $f$ by $\mathcal{A}$. For any PPT adversary $\mathcal{A}$, there exists a PPT algorithm $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\text{1-ABE}}_{\mathcal{A}}(\lambda) := |\Pr[\langle \mathcal{A}, \mathsf{G}^{\text{1-ABE}}_0\rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{G}^{\text{1-ABE}}_1\rangle = 1]|$$
$$\le 2^{9B+2}(\mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda) + 2^{-\Omega(\lambda)}).$$

Proof. Following the piecewise guessing framework, we define a series of selective hybrids $\widehat{\mathsf{H}}^{h_0}$ to $\widehat{\mathsf{H}}^{h_L}$, where $L = 8^B$, and two intermediate games $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}$ and $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}}$, which satisfy

- $\widehat{\mathsf{G}}^{\text{1-ABE}}_0 = \widehat{\mathsf{H}}^{h_0} \approx_c, \ldots, \approx_c \widehat{\mathsf{H}}^{h_L} = \widehat{\mathsf{G}}^{\text{1-ABE}}_{\mathsf{M0}}$
- $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}} = \mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}}$.

Let $z := (x, y) \in \{0,1\}^R$ on which $\mathcal{A}$ queries $O_X$ and $O_F$, respectively. Let $b \in \{0,1\}^n$ be a string computed from $z$ following Definition 2.7. Note that $f(b) = 0$ because the game impose the condition $R(x,y) = 0$ on $\mathcal{A}$. Let $\mathcal{R}$ be the pebbling record generated as $\mathcal{R} = (r_1, \ldots, r_L) = \mathsf{PebRec}(f, b)$ as defined in Lemma 2.13. Then, we define $h_\iota : \{0,1\}^R \to \{0,1\}^{3B}$ as $h_\iota(z) := r_\iota$. Note that $h_0$ and $h_L$ are constant functions because they specify the pebbling configurations where no pebbles on it and a pebble is placed on only the output gate, respectively.

The hybrids and intermediate games only differ in the Share algorithm in $O_F$ as follows. That is, $\widehat{\mathsf{H}}^{h_\iota}$ is the same as $\widehat{\mathsf{G}}^{\text{1-ABE}}_0$ except that $\mathsf{Share}(f, 0)$ is replaced with $\widetilde{\mathsf{Share}}(f, 0, h_\iota(z))$, which is described in Fig 3. $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}$ is the same as $\mathsf{H}^{h_L}$, and $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}}$ is the same as $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}$ except that $\widetilde{\mathsf{Share}}(f, 0, h_L(z))$ is replaced with $\widetilde{\mathsf{Share}}(f, \mu, h_L(z))$.

We prove that

- $\mathsf{G}^{\text{1-ABE}}_0 \approx_c \mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}$,
- $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}} = \mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}}$,
- $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}} \approx_c \mathsf{G}^{\text{1-ABE}}_1$.

First, we prove item 2, then prove item 1. We omit the proof of item 3 because it is almost the same as that of item 1. Then, we are done.

$\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}} = \mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}}$. Recall that the difference between the two games lies in the input of $\widetilde{\mathsf{Share}}$, namely, $(f, 0, h_L(z))$ or $(f, \mu, h_L(z))$. First, we note that $u = h_L(z)$ is a constant that specifies the pebbling configuration on $f$ where a pebble is placed on only the output gate.

---

$\mathsf{G}^{\text{1-ABE}}_\beta$

$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mu \leftarrow \mathbb{Z}_p$, $\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k$, $\mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$, $L := \emptyset$

$\beta' \leftarrow \mathcal{A}^{O_X(\cdot), O_F(\cdot), O_R(\cdot)}(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{k})$

---

$\underline{O_X(\cdot)}$

Input: $x = (\mathbf{x} \in \mathbb{Z}_p^m, \phi) \in \mathcal{X}$

$A_0 := \mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$

For $i \in [m]$:

  If $(\phi(i), *, *) \notin L$:

    $\mathbf{W}_{\phi(i),0}, \mathbf{W}_{\phi(i),1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$

    $L := L \cup (\phi(i), \mathbf{W}_{\phi(i),0}, \mathbf{W}_{\phi(i),1})$

  $A_i := (x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{c}$

Output $(A_0, \{A_i\}_{i\in[m]})$

---

$\underline{O_F(\cdot)}$

Input: $y = (\mathbf{y} \in \mathbb{Z}_p^n, f, \psi, t) \in \mathcal{Y}$

$\mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k})$, $\sigma_1, \ldots, \sigma_n \leftarrow \boxed{\mathsf{Share}(f, \beta\mu)}$

$\pi(i) := |\{j \mid \psi(j) = \psi(i), j \le i\}|$

$d := \max_{i\in[n]} \pi(i)$

$\mathbf{r}_1, \ldots, \mathbf{r}_d \leftarrow \mathbb{Z}_p^k$

$P_0 := ([\mathbf{B}\mathbf{r}_1]_2, \ldots, [\mathbf{B}\mathbf{r}_d]_2)$

For $i \in [n]$:

  If $(\psi(i), *, *) \notin L$:

    $\mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$

    $L := L \cup (\psi(i), \mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1})$

  If $t(i) = 1$ :

    $P_i := [\mathbf{k}_i + \sigma_i \mathbf{a}^\perp + (y_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1$

  Else:

    $P_i := \begin{pmatrix} [-(\mathbf{k}_i + \sigma_i\mathbf{a}^\perp) + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [y_i(\mathbf{k}_i + \sigma_i\mathbf{a}^\perp) + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix}$

Output $(P_0, \{P_i\}_{i\in[n]})$

---

$\underline{O_R(\cdot)}$

Input: $i \in \{0,1\}^*$

If $(i, *, *) \notin L$:

  $\mathbf{W}_{i,0}, \mathbf{W}_{i,1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$, $L := L \cup (i, \mathbf{W}_{i,0}, \mathbf{W}_{i,1})$

Output $(\mathbf{W}^\top_{i,0}\mathbf{A}, \mathbf{W}^\top_{i,1}\mathbf{A})$

**Figure 2: Core 1-ABE game.**

$\widetilde{\mathsf{Share}}(f, \mathbf{k}, u)$

Input: $f : \{0,1\}^n \to \{0,1\}$ with a depth $B$, $\mathbf{k} \in \mathbb{Z}_p^\ell$, and $u \in \{0,1\}^{3B}$

(1) Set a vector $\boldsymbol{\sigma}_{\mathsf{out}} := \mathbf{k}$ on the output wire.
(2) Interpret $u$ as a pebbling configuration on $f$.
(3) For each gate $g$ with a pebble that has incoming wires $a, b$ and an outgoing wire $c$ where a vector $\boldsymbol{\sigma}_c$ is set on $c$, choose $\mathbf{u}_{g,1}, \mathbf{u}_{g,2} \leftarrow \mathbb{Z}_p^\ell$ and set $\boldsymbol{\sigma}_a := \mathbf{u}_{g,1}$ and $\boldsymbol{\sigma}_b := \mathbf{u}_{g,2}$ on $a$ and $b$, respectively.
(4) For each AND gate $g$ with no pebble that has incoming wires $a, b$ and an outgoing wire $c$ where a vector $\boldsymbol{\sigma}_c$ is set on $c$, choose $\mathbf{u}_g \leftarrow \mathbb{Z}_p^\ell$ and set $\boldsymbol{\sigma}_a := \boldsymbol{\sigma}_c - \mathbf{u}_g$ and $\boldsymbol{\sigma}_b := \mathbf{u}_g$ on $a$ and $b$, respectively.
(5) For each OR gate $g$ with no pebble that has incoming wires $a, b$ and an outgoing wire $c$ where a vector $\boldsymbol{\sigma}_c$ is set on $c$, set $\boldsymbol{\sigma}_a := \boldsymbol{\sigma}_c$ and $\boldsymbol{\sigma}_b := \boldsymbol{\sigma}_c$ on $a$ and $b$, respectively.
(6) For each input wire $i$ with a pebble, replace $\boldsymbol{\sigma}_i$ with a random vector $\mathbf{u}_i \leftarrow \mathbb{Z}_p^k$.
(7) Output shares $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n$, which are set on the input wires $1, \ldots, n$.

**Figure 3: Description of $\widetilde{\mathsf{Share}}$.**

In this case, it is not difficult to see that the output of $\widetilde{\mathsf{Share}}$ is independent of the second argument of the input. This is because the values set on the two incoming wires of the output gate are chosen independently of $\sigma_{\mathsf{out}}$ when a pebble is placed on the output gate (see item 3 in Fig 3). Then, the values to be set on the rest of wires are computed based on these values set on the incoming wires of the output gate. Thus, the output of $\widetilde{\mathsf{Share}}$ is identically distributed in both games, and the claim holds.

$\mathsf{G}_0^{\text{1-ABE}} \approx_c \mathsf{G}_{\mathsf{M0}}^{\text{1-ABE}}$. Following Lemma 2.10, we prove the two properties:

(1) $\mathsf{G}_0^{\text{1-ABE}} = \mathsf{H}^{h_0}$ and $\mathsf{H}^{h_L} = \mathsf{G}_{\mathsf{M0}}^{\text{1-ABE}}$,
(2) $\widehat{\mathsf{H}}_1^{h_{\iota-1}} \approx_c \widehat{\mathsf{H}}_0^{h_\iota}$ for $\iota \in [L]$.

where $\widehat{\mathsf{H}}_\beta^{h_i}$ for $\beta \in \{0,1\}$ is defined in Section 2.4. For item 1, the latter holds because we defined $\mathsf{G}_{\mathsf{M0}}^{\text{1-ABE}}$ in such a way. To show the former, we need to confirm that the output of $\mathsf{Share}(f, 0)$ and $\widetilde{\mathsf{Share}}(f, 0, h_0(z))$ is identically distributed. Recall that $h_0$ is a constant function that specifies the pebbling configuration where no pebbles on it. In this case, no gates correspond to item 3 or 6 in Fig 3, and the remaining procedures are exactly the same as $\mathsf{Share}(f, 0)$. Thus, the former also holds.

The remaining thing is to prove $\widehat{\mathsf{H}}_1^{h_{\iota-1}} \approx_c \widehat{\mathsf{H}}_0^{h_\iota}$. Formally, we show that, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_1^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_0^{h_\iota} \rangle = 1]| \leq 2\mathsf{Adv}_{\mathcal{B},\mathsf{bi}}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

To show this, we additionally consider three intermediate selective hybrids $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ to $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$.

In the following, we denote the pebbling configuration on $f$ that is specified by a bit string $u$ by $C(f, u)$. Let $u_0$ and $u_1$ be the committed values by $\mathcal{A}$, which correspond to $h_{\iota-1}(z)$ and $h_\iota(z)$ for $z$ chosen by $\mathcal{A}$. Then, $C(f, u_0)$ and $C(f, u_1)$ are adjacent pebbling configurations for some input $b \in \{0,1\}^n$ for $f$. In other words, there exists $b$ such that $u_0$ and $u_1$ correspond to $r_{\iota-1}$ and $r_\iota$ where $(r_0, \ldots, r_L) = \mathsf{PebRec}(f, b)$. Thus, $C(f, u_0)$ can be changed to $C(f, u_1)$ in one step following the rule defined in Definition 2.11. Recall that the difference between $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ is the input of $\widetilde{\mathsf{Share}}$. That is, the input is $(f, 0, u_0)$ in $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $(f, 0, u_1)$ in $\widehat{\mathsf{H}}_0^{h_\iota}$. Thus, in case of $u_0 = u_1$, $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are clearly identical. In the following, we consider the case of $u_0 \neq u_1$.

Let an object $O$ be either a gate $g$ with incoming wires $a, b$ and an outgoing $c$ or an input wire $i^*$, in which the difference between $C(f, u_0)$ and $C(f, u_1)$ lies. We consider only the case where a pebble is placed on $g$ or $i^*$, since the case where a pebble is removed is just the reverse of the former case. Intermediate hybrids $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ to $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ are different from $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ only in $O_F$ as shown in Fig 4. That is, when $O$ is a gate, $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ to $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ are the same as $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$. When $O$ is an input wire, these hybrids are defined as follows:

- $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ is the same as $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ except that $\mathbf{v}_{\pi(i^*)} := \mathbf{d}$ for $\mathbf{d} \leftarrow \mathbb{Z}_p^{k+1}$,
- $\widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}}$ is the same as $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ except that random value $u$ is added to $\sigma_{i^*}$,
- $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ is the same as $\widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}}$ except that $\mathbf{v}_{\pi(i^*)} := \mathbf{B}\mathbf{r}_{\pi(i^*)}$ for $\mathbf{r}_{\pi(i^*)} \leftarrow \mathbb{Z}_p^k$.

Thanks to Lemmas 3.4 to 3.7 and observations so far, Lemma 3.3 holds. □

Lemma 3.4.

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_1^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}} \rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B},\mathsf{bi}}^{\mathcal{D}_k\text{-MDDH}}(\lambda).$$

Proof. We show that the $\mathcal{D}_k$-MDDH problem is reduced to this difference. The reduction algorithm $\mathcal{B}$ is given an instance $(\mathbb{G}, [\mathbf{B}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$ where $\mathbf{t}_0 = \mathbf{B}\mathbf{r}$ and $\mathbf{t}_1 = \mathbf{d}$. $\mathcal{B}$ gives $[\mathbf{B}]_{1,2}$ to $\mathcal{A}$ as its input ($\mathcal{B}$ generates other inputs by itself). When $\mathcal{A}$ queries $O_X$ and $O_R$, $\mathcal{B}$ replies honestly. When $\mathcal{A}$ queries $O_F$, $\mathcal{B}$ replies honestly except that it sets $[\mathbf{v}_{\pi(i^*)}]_{1,2} := [\mathbf{t}_\beta]_{1,2}$. Then the view of $\mathcal{A}$ corresponds to $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ if $\beta = 0$, and $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ otherwise. This concludes the proof. □

Lemma 3.5.

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}} \rangle = 1] \leq 2^{-\Omega(\lambda)}.$$

Proof. We redefine that $\mathbf{W}_{\psi(i^*),b} := \widetilde{\mathbf{W}}_{\psi(i^*),b} + w_{\psi(i^*),b}\mathbf{a}^\perp\mathbf{b}^{\perp\top}$, where $\widetilde{\mathbf{W}}_{\psi(i^*),b} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$, $w_{\psi(i^*),b} \leftarrow \mathbb{Z}_p$, and $b \in \{0,1\}$. Since $\widetilde{\mathbf{W}}_{\psi(i^*),b}$ is chosen randomly, the distribution of redefined $\mathbf{W}_{\psi(i^*),b}$ is identical to that of the original definition. Observe that this change does not affect the outputs of $O_R$ because $\mathbf{a}^{\perp\top}\mathbf{A} = \mathbf{0}^\top$.

$\widehat{\mathsf{H}}_1^{h_{\iota-1}}$, $\boxed{\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}}$, $\boxed{\widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}}}$, $\boxed{\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}}$

$\underline{O_F(\cdot)}$

Input: $y = (\mathbf{y} \in \mathbb{Z}_p^n, f, \psi, t) \in \mathcal{Y}$

$\mathbf{k}_1, \dots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k}),\ \ \sigma_1, \dots, \sigma_n \leftarrow \widetilde{\mathsf{Share}}(f, 0, u_0)$

$\pi(i) := |\{j \mid \psi(j) = \psi(i), j \le i\}|$

$d := \max_{i \in [n]} \pi(i)$

$\mathbf{r}_1, \dots, \mathbf{r}_d \leftarrow \mathbb{Z}_p^k,\ \ \mathbf{d} \leftarrow \mathbb{Z}_p^{k+1}$

$\mathbf{v}_i := \mathbf{B}\mathbf{r}_i$ for $i \in [d]$

$\boxed{\mathbf{v}_i := \mathbf{B}\mathbf{r}_i \text{ for } i \in [d]\backslash\pi(i^*),\ \ \mathbf{v}_{\pi(i^*)} := \mathbf{d}}$

$P_0 := ([\mathbf{v}_1]_2, \dots, [\mathbf{v}_d]_2)$

For $i \in [n]$:

  If $(\psi(i), *, *) \notin L$:

    $\mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$

    $L := L \cup (\psi(i), \mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1})$

  If $i = i^*$

    $\boxed{u \leftarrow \mathbb{Z}_p,\ \ \sigma_i := \sigma_i + u}$

  If $t(i) = 1$ :

    $P_i := [\mathbf{k}_i + \sigma_i \mathbf{a}^\perp + (y_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{v}_{\pi(i)}]_1$

  Else:

    $P_i := \begin{pmatrix} [-(\mathbf{k}_i + \sigma_i \mathbf{a}^\perp) + \mathbf{W}_{\psi(i),0}\mathbf{v}_{\pi(i)}]_1, \\ [y_i(\mathbf{k}_i + \sigma_i \mathbf{a}^\perp) + \mathbf{W}_{\psi(i),1}\mathbf{v}_{\pi(i)}]_1 \end{pmatrix}$

Output $(P_0, \{P_i\}_{i \in [n]})$

**Figure 4: Description of $O_F$ in hybrids.**

For $O_F$, $P_i$ for $i \in \psi^{-1}(\psi(i^*))$ can be written as

If $t(i) = 1$ :

$$P_i := \begin{bmatrix} \mathbf{k}_i + \sigma_i \mathbf{a}^\perp + (y_i \widetilde{\mathbf{W}}_{\psi(i^*),0} + \widetilde{\mathbf{W}}_{\psi(i^*),1})\mathbf{v}_{\pi(i)} \\ +(y_i w_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{a}^\perp \mathbf{b}^{\perp\top} \mathbf{v}_{\pi(i)} \end{bmatrix}_1$$

Else:

$$P_i := \begin{pmatrix} [-(\mathbf{k}_i + \sigma_i \mathbf{a}^\perp) + \widetilde{\mathbf{W}}_{\psi(i^*),0}\mathbf{v}_{\pi(i)} + w_{\psi(i^*),0}\mathbf{a}^\perp \mathbf{b}^{\perp\top} \mathbf{v}_{\pi(i)}]_1, \\ [y_i(\mathbf{k}_i + \sigma_i \mathbf{a}^\perp) + \widetilde{\mathbf{W}}_{\psi(i^*),1}\mathbf{v}_{\pi(i)} + w_{\psi(i^*),1}\mathbf{a}^\perp \mathbf{b}^{\perp\top} \mathbf{v}_{\pi(i)}]_1 \end{pmatrix}.$$

For $i \neq i^*$, we have $\mathbf{b}^{\perp\top}\mathbf{v}_{\pi(i)} = \mathbf{b}^{\perp\top}\mathbf{B}\mathbf{r}_{\pi(i)} = 0$, and thus the distribution is not changed. For $i = i^*$, we have $\mathbf{b}^{\perp\top}\mathbf{v}_{\pi(i^*)} = \mathbf{b}^{\perp\top}\mathbf{d} \neq 0$ with overwhelming probability because $\mathbf{d}$ is chosen randomly from $\mathbb{Z}_p^{k+1}$.

Then, we consider the two cases.

- $t(i^*) = 1$. This case means that $\mathcal{A}$ either does not obtain an information on $\mathbf{W}_{\psi(i^*),b}$ or obtains a vector

$$(x\mathbf{W}_{\psi(i^*),0}^\top + \mathbf{W}_{\psi(i^*),1}^\top)\mathbf{c}$$
$$=(x\widetilde{\mathbf{W}}_{\psi(i^*),0}^\top + \widetilde{\mathbf{W}}_{\psi(i^*),1}^\top)\mathbf{c} + (x w_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}^\perp \mathbf{a}^{\perp\top}\mathbf{c}$$

  for some $x \neq y_{i^*}$ from $O_X$. In both cases, the value $(y_i w_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}^{\perp\top}\mathbf{d}$ in $P_{i^*}$ is randomly distributed from the viewpoint of $\mathcal{A}$ because this is a pairwise independent function. Thus, adding $u\mathbf{a}^\perp$ to $P_{i^*}$ does not change the distribution.

- $t(i^*) = 0$. This case means that $\mathcal{A}$ either does not obtain an information on $\mathbf{W}_{\psi(i^*),b}$ or obtains a vector

$$(x\mathbf{W}_{\psi(i^*),0}^\top + \mathbf{W}_{\psi(i^*),1}^\top)\mathbf{c}$$
$$=(x\widetilde{\mathbf{W}}_{\psi(i^*),0}^\top + \widetilde{\mathbf{W}}_{\psi(i^*),1}^\top)\mathbf{c} + (x w_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}^\perp \mathbf{a}^{\perp\top}\mathbf{c}$$

  for $x = y_{i^*}$. In both cases, setting $w_{\psi(i^*),0} := w'_{\psi(i^*),0} - u/\mathbf{b}^{\perp\top}\mathbf{d}$ and $w_{\psi(i^*),1} := w'_{\psi(i^*),1} + y_i u/\mathbf{b}^{\perp\top}\mathbf{d}$ for randomly chosen $w'_{\psi(i^*),b}$ does not change the distribution.

Thus, the views of $\mathcal{A}$ in both hybrids are identical unless $\mathbf{b}^{\perp\top}\mathbf{d} = 0$. □

Lemma 3.6.

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}} \rangle = 1]| \le \mathsf{Adv}_{\mathcal{B},\mathsf{bi}}^{\mathcal{D}_k\text{-MDDH}}(\lambda).$$

We omit the proof because the proof of this lemma is almost the same as Lemma 3.4.

Lemma 3.7.

$$\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}} \rangle = 1] = \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_0^{h_\iota} \rangle = 1].$$

Proof. In $\widehat{\mathsf{H}}_0^{h_\iota}$, the third input of $\widetilde{\mathsf{Share}}$ is changed to $u_1$ instead of $u_0$, and a random value is no longer added to $\sigma_{i^*}$ even if $O$ is an input wire $i^*$. To see that both hybrids are identical, we consider the three cases.

(1) The object $O$ is an AND gate $g$ with incoming wires $a, b$ and an outgoing wire $c$, and at least one of its incoming wires comes from a gate or input wire with a pebble, say $O'$. In this case, the outputs of $\widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\widetilde{\mathsf{Share}}(f, 0, u_1)$ are identically distributed. Wlog, we can assume that the wire $a$ comes from $O'$. The difference between $\widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\widetilde{\mathsf{Share}}(f, 0, u_1)$ is whether $\sigma_a := \sigma_c - \sigma_b$ or $\sigma_a := u$ where $u \leftarrow \mathbb{Z}_p$ is set on the wire $a$. The crucial fact is that $O'$ is independent of $\sigma_a$. That is, if $O'$ is a gate $g'$ with a pebble, the values set to its incoming wires are independent of $\sigma_a$ (see item 3 in Fig 3). If $O'$ is an input wire $i'$ with a pebble, the value set to the input wire is independent of $\sigma_a$ (see item 6 in Fig 3). Thus, $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are identical in this case.

(2) The object $O$ is an OR gate $g$ and both of its incoming wires come from a gate or an input wire with a pebble, respectively. From a similar observation to the above case, we can see that $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are identical in this case.

(3) The entity $O$ is an input wire $i^*$. Let $a'$ be an incoming wire of a gate $g'$ that the input wire $i^*$ leads to. In this case, the difference between $\widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\widetilde{\mathsf{Share}}(f, 0, u_1)$ is whether $\sigma_{i^*}$ is set to $\sigma_{a'}$ or replaced with a random value. Observe that computing $\sigma_{i^*} := \sigma_{a'} + u$ for $u \leftarrow \mathbb{Z}_p$ is the same as replacing it with a random value. Thus, $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are also identical in this case.

In conclusion, the difference between $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ is fairly conceptual. □

# 4 OUR KP-ABE SCHEME

## 4.1 Construction

Let $H : \{0,1\}^* \to G_1^{(k+1)\times k} \times G_1^{(k+1)\times k}$ be a hash function modeled as a random oracle. Let $F_K : \{0,1\}^* \to \mathbb{Z}_p^{k+1} \times \mathbb{Z}_p^{k+1}$ be a PRF with a secret key $K$. Let $\mathcal{K}_\lambda$ be a key space of the PRF. Let Share be the LSSS described in Fig 1. Then, our scheme can be described as Fig 5. Our scheme essentially subsumes the KP-ABE scheme by Agrawal and Chase [1]. This is because if we only allow a zero vector for $\mathbf{x}$ and $\mathbf{y}$ and policies with the one-use restriction, our scheme is almost the same as their scheme. Note that when we implement $H$ in practice, we can use a hash function $H' : \{0,1\}^* \to G_1$ and append a location of an element to the input to generate each element of the matrices.

For generality, we describe our scheme using a matrix distribution $\mathcal{D}_k$. When we instantiate our scheme from asymmetric pairings, we typically choose the $k$-Lin family $\mathcal{L}_k$ with $k = 2$. In this case, we can set matrices as

$$\mathbf{A} = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{A}^* = \begin{pmatrix} \frac{1}{a_1} & 0 \\ 0 & \frac{1}{a_2} \\ 0 & 0 \end{pmatrix}, \quad \mathbf{a}^\perp = \begin{pmatrix} -\frac{1}{a_1} \\ -\frac{1}{a_2} \\ 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_1 & 0 \\ 0 & b_2 \\ 1 & 1 \end{pmatrix},$$

where $a_1, a_2, b_1, b_2 \leftarrow \mathbb{Z}_p$.

## 4.2 Security

Theorem 4.1. *Let $B$ be the maximum depth of formulae on which $\mathcal{A}$ queries* KeyGen. *Let $q_{\mathsf{sk}}$ be the maximum number of $\mathcal{A}$'s queries to* KeyGen. *Then, our scheme is adaptively secure as long as $B = O(\log \lambda)$. More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_1}(\lambda) + (2^{9B+2} q_{\mathsf{sk}} + 1)(\mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B}_2,\mathsf{bi}}(\lambda) + 2^{-\Omega(\lambda)}).$$

**Proof overview.** We prove the lemma following the standard dual system methodology. To do so, we first replace the PRF with a random function. Then, our scheme basically follows the construction on the dual system group from prime-order groups in [11]. Concretely, we can rewrite $c_{2,i}$ and $k_{2,i}$ in the challenge ciphertext and secret keys as

$$\begin{aligned} c_{2,i} &= [(x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1, \\ k_{2,i} &:= [\mathbf{k}_i + (y_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \quad \text{if } t(i) = 1, \\ k_{2,i} &:= \begin{pmatrix} [-\mathbf{k}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [y_i \mathbf{k}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix} \quad \text{if } t(i) = 0, \end{aligned}$$

where $\mathbf{W}_{i,b} \in \mathbb{Z}_p^{(k+1)\times(k+1)}$. Next, we change the challenge ciphertext into a semi-functional form, where term $\mathbf{As}$ is replaced with a vector $\mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$. That is, the elements in a ciphertext are

$$c_1 = [\mathbf{c}]_2,\ c_{2,i} = [(x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{c}]_1,\ c_3 = [\mathbf{c}^\top \mathbf{k}]_T M.$$

The indistinguishability directly follows from the $\mathcal{D}_k$-MDDH assumption. After that, we gradually change the secret keys into a semi-functional form, where $\mathbf{k}_i$ is a share of secret $\mathbf{k} + \mu \mathbf{a}^\top$ instead of $\mathbf{k}$. To prove each indistinguishability, we utilize Lemmas 3.1 and 3.3. In the final hybrid, we can argue that $\mathbf{c}^\top \mathbf{k}$ in the challenge ciphertext is statistically close to an uniform randomness.

Proof. We consider a series of hybrids $\mathsf{H}_0$, $\mathsf{H}_1$, $\mathsf{H}_2$, and $\mathsf{H}_{3,\iota}$ for $i \in \{0, \dots, q_{\mathsf{sk}}\}$, where $\mathsf{H}_0$ is the real game and $\mathsf{H}_{3,q_{\mathsf{sk}}}$ is the final game. In the following, we denote the event $\beta = \beta'$ in hybrid $\mathsf{H}$ by $\langle \mathcal{A}, \mathsf{H} \rangle_{\mathsf{win}}$, where $\beta$ is a random bit chosen by challenger $C$, and $\beta'$ is the output of $\mathcal{A}$. Note that we have

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_0 \rangle_{\mathsf{win}}] - 1/2| = \mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda). \tag{1}$$

$\underline{\mathsf{H}_1}$. We define $\mathsf{H}_1$ as the same as $\mathsf{H}_0$ except replacing PRF $F_K$ in KeyGen with a random function $R : \{0,1\}^* \to \mathbb{Z}_p^{k+1} \times \mathbb{Z}_p^{k+1}$. From the definition of PRFs, we have

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_0 \rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_1 \rangle_{\mathsf{win}}]| \le \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda). \tag{2}$$

$\underline{\mathsf{H}_2}$. Next, we define $\mathsf{H}_2$. We change the behavior of random oracle $H$ and random function $R$. Consider another random oracle $H' : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times(k+1)} \times \mathbb{Z}_p^{(k+1)\times(k+1)}$ that only challenger $C$ can access. We denote the first and second elements of $H'(i)$ by $\mathbf{W}_{i,0}$ and $\mathbf{W}_{i,1}$, respectively. In $\mathsf{H}_2$, $H(i)$ outputs $([\mathbf{W}^\top_{i,0}\mathbf{A}]_1, [\mathbf{W}^\top_{i,1}\mathbf{A}]_1)$, and $R(i)$ outputs $(\mathbf{W}^\top_{i,0}\mathbf{a}_{\mathsf{R}}, \mathbf{W}^\top_{i,1}\mathbf{a}_{\mathsf{R}})$. Then, we have

$$\Pr[\langle \mathcal{A}, \mathsf{H}_1 \rangle_{\mathsf{win}}] = \Pr[\langle \mathcal{A}, \mathsf{H}_2 \rangle_{\mathsf{win}}]. \tag{3}$$

It is not difficult to confirm that the above equality holds because $\overline{\mathbf{A}} = (\mathbf{A} || \mathbf{a}_{\mathsf{R}})$ is a regular matrix, and thus $\mathbf{W}^\top_{i,b}\overline{\mathbf{A}}$ is randomly distributed in $\mathbb{Z}_p^{(k+1)\times(k+1)}$ for $\mathcal{A}$. By this conceptual change, we can rewrite $c_{2,i}$ and $k_{2,i}$ in the challenge ciphertext and secret keys as follows:

$$\begin{aligned} c_{2,i} &= [(x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1, \\ k_{2,i} &:= [\mathbf{k}_i + (y_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \quad \text{if } t(i) = 1, \\ k_{2,i} &:= \begin{pmatrix} [-\mathbf{k}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [y_i \mathbf{k}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix} \quad \text{if } t(i) = 0 \end{aligned}$$

In the above, we use the relations $\mathbf{A}^*\mathbf{A}^\top + \mathbf{a}^\perp \mathbf{a}^\top_{\mathsf{R}} = \mathbf{I}_{k+1}$.

$\underline{\mathsf{H}_{3,\iota}}$. To describe $\mathsf{H}_{3,\iota}$, we define some distributions on ciphertexts and secret keys as follows. Concretely, we define two types of ciphertexts and secret keys, namely, normal and semi-functional. A normal ciphertext is one generated as in $\mathsf{H}_2$. That is,

$$c_1 = [\mathbf{As}]_2,\ c_{2,i} = [(x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1,\ c_3 = [\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}]_T M.$$

A semi-functional ciphertext is the same as the normal one except that $\mathbf{As}$ is replaced with $\mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$. That is,

$$c_1 = [\mathbf{c}]_2,\ c_{2,i} = [(x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{c}]_1,\ c_3 = [\mathbf{c}^\top \mathbf{k}]_T M.$$

Similarly, a normal secret key is one generated as in $\mathsf{H}_2$. That is,

$$\begin{aligned} k_{1,j} &= [\mathbf{B}\mathbf{r}_j]_2, \\ k_{2,i} &:= [\mathbf{k}_i + (y_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \quad \text{if } t(i) = 1, \\ k_{2,i} &:= \begin{pmatrix} [-\mathbf{k}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [y_i \mathbf{k}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix} \quad \text{if } t(i) = 0 \end{aligned} \tag{4}$$

Especially, $\mathbf{k}_1, \dots, \mathbf{k}_n$ in $k_{2,i}$ is outputs of $\mathsf{Share}(f, \mathbf{k})$. On the other hand, in a semi-functional secret key, $\mathbf{k}_1, \dots, \mathbf{k}_n$ in $k_{2,i}$ is outputs of $\mathsf{Share}(f, \mathbf{k} + \mu \mathbf{a}^\perp)$ where $\mu \leftarrow \mathbb{Z}_p$. Then, $\mathsf{H}_{3,\iota}$ is the same as $\mathsf{H}_2$ except that the challenge ciphertext and the first $\iota$ keys that $\mathcal{A}$ is given are semi-functional.

$\mathsf{Setup}(1^\lambda)$**:** It takes a security parameter $1^\lambda$ and outputs pk and msk as follows.

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda),\ \ \mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k,\ \ \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1},\ \ K \leftarrow \mathcal{K}_\lambda,$$
$$\mathsf{pk} := (\mathbb{G}, [\mathbf{A}]_2, [\mathbf{A}^\top \mathbf{k}]_T),\ \ \mathsf{msk} := (\mathbf{A}^*, \mathbf{a}^\perp, \mathbf{B}, \mathbf{k}, K).$$

$\mathsf{Enc}(\mathsf{pk}, x, M)$**:** It takes pk, an attribute $x = (\mathbf{x} \in \mathbb{Z}_p^m, \phi)$, and a message $M \in G_T$ and outputs $\mathsf{ct}_x$ as follows.

$$\mathbf{s} \leftarrow \mathbb{Z}_p^k,\ \ ([\mathbf{U}_{\phi(i),0}]_1, [\mathbf{U}_{\phi(i),1}]_1) := H(\phi(i)),$$
$$c_1 := [\mathbf{A}\mathbf{s}]_2,\ \ c_{2,i} := [(x_i \mathbf{U}_{\phi(i),0} + \mathbf{U}_{\phi(i),1})\mathbf{s}]_1,\ \ c_3 := [\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}]_T M \ \text{ for } i \in [m],$$
$$\mathsf{ct}_x := (x, c_1, \{c_{2,i}\}_{i\in[m]}, c_3).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$**:** It takes pk, msk, and a predicate $y = (\mathbf{y} \in \mathbb{Z}_p^n, f, \psi, t)$ and outputs $\mathsf{sk}_y$ as follows. Let $\pi : [n] \to \mathbb{N}$ be a function such that $\pi(i) := |\{j \mid \psi(j) = \psi(i), j \le i\}|$. Let $d$ be the maximum number of multi-use of labels in $f$, i.e., $d := \max_{i\in[n]} \pi(i)$.

$$\mathbf{r}_1, \dots, \mathbf{r}_d \leftarrow \mathbb{Z}_p^k,\ \ \mathbf{k}_1, \dots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k}) \in \mathbb{Z}_p^{k+1},$$
$$k_{1,j} := [\mathbf{B}\mathbf{r}_j]_2 \ \text{ for } j \in [d],$$
$$([\mathbf{U}_{\psi(i),0}]_1, [\mathbf{U}_{\psi(i),1}]_1) := H(\psi(i)),\ \ (\mathbf{u}_{\psi(i),0}, \mathbf{u}_{\psi(i),1}) := F_K(\psi(i)),$$
$$k_{2,i} := [\mathbf{k}_i + \mathbf{A}^*(y_i \mathbf{U}_{\psi(i),0}^\top + \mathbf{U}_{\psi(i),1}^\top)\mathbf{B}\mathbf{r}_{\pi(i)} + \mathbf{a}^\perp(y_i \mathbf{u}_{\psi(i),0}^\top + \mathbf{u}_{\psi(i),1}^\top)\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1,$$
$$k_{2,i} := (k_{2,i,1}, k_{2,i,2}) := \begin{pmatrix} [-\mathbf{k}_i + \mathbf{A}^*\mathbf{U}_{\psi(i),0}^\top \mathbf{B}\mathbf{r}_{\pi(i)} + \mathbf{a}^\perp \mathbf{u}_{\psi(i),0}^\top \mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [y_i \mathbf{k}_i + \mathbf{A}^*\mathbf{U}_{\psi(i),1}^\top \mathbf{B}\mathbf{r}_{\pi(i)} + \mathbf{a}^\perp \mathbf{u}_{\psi(i),1}^\top \mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix} \ \text{ if } t(i) = 0$$
$$\text{for } i \in [n],$$
$$\mathsf{sk}_y := (y, \{k_{1,j}\}_{j\in[d]}, \{k_{2,i}\}_{i\in[n]}).$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)$**:** It takes pk, $\mathsf{ct}_x$, and $\mathsf{sk}_y$. It computes $b \in \{0,1\}^n$ from $x$ and $y$ as in Definition 2.7. If $f(b) = 0$, it outputs $\perp$. Otherwise, computes a set $S \subseteq \{i \mid b_i = 1\}$ such that $\mathbf{k} = \sum_{i\in S} \mathbf{k}_i$. Let $S_1 := S \cap \{i \mid t(i) = 1\}$ and $S_0 := S \cap \{i \mid t(i) = 0\}$. Then outputs $M'$ as follows.

$$D_{1,j} := e\left( \sum_{\substack{\pi(i)=j \\ i\in S_1}} k_{2,i} + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \frac{1}{y_i - x_{\phi^{-1}(\psi(i))}} (x_{\phi^{-1}(\psi(i))} k_{2,i,1} + k_{2,i,2}), c_1 \right)^\top,$$
$$D_{2,j} := e\left( \sum_{\substack{\pi(i)=j \\ i\in S_1}} c_{2,\phi^{-1}(\psi(i))} + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \frac{1}{y_i - x_{\phi^{-1}(\psi(i))}} c_{2,\phi^{-1}(\psi(i))}, k_{1,j} \right) \ \text{ for } j \in [d],$$
$$M' := c_3 / \prod_{j\in[d]} (D_{1,j}/D_{2,j}).$$

**Correctness:** For honestly generated $\mathsf{ct}_x$ and $\mathsf{sk}_y$ such that $R(x,y) = 1$, we have

$$D_{1,j} = \left[ \sum_{\substack{\pi(i)=j \\ i\in S_1}} \left( \mathbf{s}^\top \mathbf{A}^\top \mathbf{k}_i + \mathbf{s}^\top (y_i \mathbf{U}_{\psi(i),0}^\top + \mathbf{U}_{\psi(i),1}^\top)\mathbf{B}\mathbf{r}_j \right) + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \left( \mathbf{s}^\top \mathbf{A}^\top \mathbf{k}_i + \frac{1}{y_i - x_{\phi^{-1}(\psi(i))}} \mathbf{s}^\top (x_{\phi^{-1}(\psi(i))} \mathbf{U}_{\psi(i),0}^\top + \mathbf{U}_{\psi(i),1}^\top)\mathbf{B}\mathbf{r}_j \right) \right]_T \in G_T,$$
$$D_{2,j} = \left[ \sum_{\substack{\pi(i)=j \\ i\in S_1}} \left( \mathbf{s}^\top (x_{\phi^{-1}(\psi(i))} \mathbf{U}_{\psi(i),0}^\top + \mathbf{U}_{\psi(i),1}^\top)\mathbf{B}\mathbf{r}_j \right) + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \left( \frac{1}{y_i - x_{\phi^{-1}(\psi(i))}} \mathbf{s}^\top (x_{\phi^{-1}(\psi(i))} \mathbf{U}_{\psi(i),0}^\top + \mathbf{U}_{\psi(i),1}^\top)\mathbf{B}\mathbf{r}_j \right) \right]_T \in G_T.$$

In the above, we use the relations $\mathbf{A}^\top \mathbf{A}^* = \mathbf{I}_k$ and $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$. Because $x_{\phi^{-1}(\psi(i))} = y_i$ for $i \in S_1$, we have $\prod_{j\in[d]} (D_{1,j}/D_{2,j}) = [\mathbf{s}^\top \mathbf{A}^\top \sum_{j\in[d]} \sum_{\substack{i\in S \\ \pi(i)=j}} \mathbf{k}_i]_T = [\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}]_T$. Thus, $M' = M$.

**Figure 5: Our KP-ABE Scheme.**

Lemma 4.2.

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_2\rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_{3,0}\rangle_{\mathsf{win}}]| \le \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda). \quad (5)$$

Proof. To show this, we describe $\mathcal{B}$, which is given an instance of the $\mathcal{D}_k$-MDDH problem $(\mathbb{G}, [\mathbf{A}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$.

(1) $\mathcal{B}$ generates $\mathbf{B}$ and $\mathbf{k}$ by itself.
(2) $\mathcal{B}$ computes $\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_2, e([\mathbf{A}]_1, [\mathbf{k}]_2))$ and gives it to $\mathcal{A}$.
(3) For query $H(i)$, $\mathcal{B}$ answers with $([\mathbf{W}^\top_{i,0}\mathbf{A}]_1, [\mathbf{W}^\top_{i,1}\mathbf{A}]_1)$, where $(\mathbf{W}_{i,0}, \mathbf{W}_{i,1})$ is an output of $H'(i)$.
(4) For query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes $\mathsf{sk}_y$ as in Eq.(4). Note that $\mathcal{B}$ can generate sk without the random function $R$ because it does not contain terms related to $\mathbf{A}$ any more.
(5) For the challenge query with the attribute $x^* = (\mathbf{x}, \phi)$, $\mathcal{B}$ flip the coin $\delta \leftarrow \{0,1\}$ and generates $\mathsf{ct}_{x^*}$ as

$$c_1 = [\mathbf{t}_\beta]_2,\ c_{2,i} = [(x_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{t}_\beta]_1,$$
$$c_3 = e([\mathbf{t}_\beta]_1, [\mathbf{k}]_2) M_\delta.$$

(6) $\mathcal{B}$ outputs $\mathsf{true}(\delta = \delta')$, where $\delta'$ is an output of $\mathcal{A}$.

Clearly, the case $\beta = 0$ corresponds to $\mathsf{H}_2$ and the case $\beta = 1$ corresponds to $\mathsf{H}_{3,0}$. □

Lemma 4.3. *For $\iota \in [q_{\mathsf{sk}}]$, we have*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota-1}\rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota}\rangle_{\mathsf{win}}]| \le \mathsf{Adv}^{\text{1-ABE}}_{\mathcal{B}}(\lambda). \quad (6)$$

Proof. We describe $\mathcal{B}$'s behavior.

(1) $\mathcal{B}$ is given $(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{k})$ from the 1-ABE game.
(2) $\mathcal{B}$ gives $\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_2, [\mathbf{A}^\top\mathbf{k}]_T)$ to $\mathcal{A}$.
(3) For query $H(i)$, $\mathcal{B}$ answers with the reply of $O_R(i)$.
(4) For the challenge query with an attribute $x^*$, $\mathcal{B}$ flip the coin $\delta \leftarrow \{0,1\}$. Then, $\mathcal{B}$ obtains $(A_0, \{A_i\}_{i\in[m]})$ as the reply of $O_X(x^*)$. $\mathcal{B}$ returns $\mathsf{ct}_{x^*}$ computing as

$$\mathsf{ct}_{x^*} := \left([A_0]_2, \{[A_i]_1\}_{i\in[m]}, [A_0^\top\mathbf{k}]_T M_\delta\right).$$

(5) For the $\ell$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\ell < \iota$ and $y = (\mathbf{y}, f, \psi, t)$, $\mathcal{B}$ compute $\mathsf{sk}_y$ as in Eq.(4) by setting $\mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k} + \mu\mathbf{a}^\perp)$ with a fresh randomness $\mu \leftarrow \mathbb{Z}_p$.
(6) For the $\ell$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\ell = \iota$ and $y = (\mathbf{y}, f, \psi, t)$, $\mathcal{B}$ obtains $(P_0, \{P_i\}_{i\in[n]})$ as the reply of $O_F(y)$. Then, $\mathcal{B}$ returns $\mathsf{sk}_y$ computing as

$$\mathsf{sk}_y := (P_0, \{P_i\}_{i\in[n]}).$$

(7) For the $\ell$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\ell > \iota$ and $y = (\mathbf{y}, f, \psi, t)$, $\mathcal{B}$ compute $\mathsf{sk}_y$ as in Eq.(4) by setting $\mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k})$.
(8) $\mathcal{B}$ outputs $\mathsf{true}(\delta = \delta')$, where $\delta'$ is an output of $\mathcal{A}$.

From Lemma 3.1, the term $\mathbf{k}_i + \sigma_i\mathbf{a}^\perp$ in the reply of $O_F$ is identically distributed with the $i$-th output of $\mathsf{Share}(\mathbf{k} + \beta\mu\mathbf{a}^\perp)$. Thus, if the oracles are those in $\mathsf{G}_0^{\text{1-ABE}}$, $\mathcal{A}$'s view corresponds to $\mathsf{H}_{3,\iota-1}$, and otherwise, it corresponds to $\mathsf{H}_{3,\iota}$. □

Lemma 4.4.

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{3,q_{\mathsf{sk}}}\rangle_{\mathsf{win}}] - 1/2| \le 2^{-\Omega(\lambda)}. \quad (7)$$

Proof. Because $(\mathbf{A}^*||\mathbf{a}^\perp)$ forms a basis, redefining $\mathbf{k}$ as $\mathbf{k} := \mathbf{A}^*\mathbf{z} + z\mathbf{a}^\perp$ where $\mathbf{z} \leftarrow \mathbb{Z}_p^k$ and $z \leftarrow \mathbb{Z}_p$ does not change its distribution. Recall that the information on $\mathbf{k}$ that $\mathcal{A}$ obtains throughout the game is $\mathbf{A}^\top\mathbf{k}$ in pk, $\mathsf{Share}(f, \mathbf{k} + \mu\mathbf{a}^\perp)$ in $\mathsf{sk}_y$, and $\mathbf{c}^\top\mathbf{k}$ in $\mathsf{ct}_{x^*}$. However, $\mathbf{A}^\top\mathbf{k}$ does not contain the information on $z$ because $\mathbf{A}^\top\mathbf{a}^\perp = \mathbf{0}$. Similarly, each $\mathbf{k} + \mu\mathbf{a}^\perp$ also does not contain the information on $z$ because it is masked by fresh randomness $\mu$. Thus, $z\mathbf{c}^\top\mathbf{a}^\perp$ is randomly distributed in $\mathbb{Z}_p$ for $\mathcal{A}$, and so is $\mathbf{c}^\top\mathbf{k}$, unless $\mathbf{c}^\top\mathbf{a}^\perp = 0$. Since $\mathbf{c}$ is randomly chosen from $\mathbb{Z}_p^{k+1}$, $\mathbf{c}^\top\mathbf{a}^\perp = 0$ with a probability $2^{-\Omega(\lambda)}$. If it is not the case, $\mathsf{ct}_{x^*}$ does not have information on $\beta$, and the lemma holds. □

Thanks to Eq.(1) to (3) and (5) to (7) and Lemma 3.3, Theorem 4.1 holds. □

# 5 OUR CP-ABE SCHEME

*Definition 5.1 (Matrix Notation).* In this section, we use additional notations for a matrix $\overline{\mathbf{B}} \in \mathrm{GL}_{k+2}(\mathbb{Z}_p)$. $\mathbf{B}$, $\mathbf{b}_1$, and $\mathbf{b}_2$ denote a matrix and vectors consist of the first $k$ columns, the $k+1$-th column, and the last column of $\overline{\mathbf{B}}$, respectively. Similarly, $\mathbf{B}^*$, $\mathbf{b}_1^*$, and $\mathbf{b}_2^*$ denote a matrix and vectors consist of the first $k$ columns, the $k+1$-th column, and the last column of $(\overline{\mathbf{B}}^\top)^{-1}$, respectively. For the convenience, we denote $(\mathbf{b}_1||\mathbf{b}_2)$ by $\mathbf{B}_{12}$, and this notation is applied to other cases similarly.

## 5.1 Construction

Let $H : \{0,1\}^* \to G_1^{(k+1)\times k} \times G_1^{(k+1)\times k}$ be a hash function modeled as a random oracle. Let $F_K : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times 2} \times \mathbb{Z}_p^{(k+1)\times 2}$ be a PRF with a secret key $K$. Let $\mathcal{K}_\lambda$ be a key space of the PRF. Let Share be the LSSS described in Fig 1. Then, our scheme can be described as Fig 6.

For generality, we describe our scheme using a parameter $k$ and distribution $\mathcal{D}_k$. Similarly to our KP-ABE scheme, when we instantiate our scheme from asymmetric pairings, we can choose the $k$-Lin family $\mathcal{L}_k$ with $k = 2$.

## 5.2 Security

Theorem 5.2. *Let $B$ be the maximum depth of formulae for the challenge ciphertext. Let $q_{\mathsf{sk}}$ be the maximum number of $\mathcal{A}$'s queries to* KeyGen. *Then, our scheme is adaptively secure as long as $B = O(\log\lambda)$. More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_1}(\lambda) + ((2^{9B+2}+2)q_{\mathsf{sk}} + 1)(\mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B}_2,\mathsf{bi}}(\lambda) + 2^{-\Omega(\lambda)}).$$

**Proof overview.** Although the security proof of our CP-ABE scheme also follows the dual system methodology and KW framework [21], it is more complicated than the proof of our KP-ABE scheme. The main reason arises from the fact that we need a kind of sub-group assumption in the proof of the core 1-ABE indistinguishability in contrast to the KW19 framework. In the dual system methodology, we first change the challenge ciphertext into the semi-functional one and then gradually change secret keys into

$\mathsf{Setup}(1^\lambda)$**:** It takes a security parameter $1^\lambda$ and outputs pk and msk as follows.

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda),\ \ \mathbf{A} \leftarrow \mathcal{D}_k,\ \ \overline{\mathbf{B}} \leftarrow \mathsf{GL}_{k+2}(\mathbb{Z}_p),\ \ \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)},\ \ \mathbf{k} \leftarrow \mathbb{Z}_p^{k+2},\ \ K \leftarrow \mathcal{K}_\lambda,$$
$$\mathsf{pk} := (\mathbb{G}, [\mathbf{B}]_2, [\mathbf{WB}]_1, [\mathbf{B}^\top \mathbf{k}]_T),\ \ \mathsf{msk} := (\mathbf{A}, \mathbf{W}^\top \mathbf{A}, \mathbf{B}^*, \mathbf{B}^*_{12}, \mathbf{k}, K).$$

$\mathsf{Enc}(\mathsf{pk}, x, M)$**:** It takes pk, an attribute $x = (\mathbf{x} \in \mathbb{Z}_p^n, f, \psi, t)$, and a message $M \in G_T$ and outputs $\mathsf{ct}_x$ as follows. Let $\pi : [n] \to \mathbb{N}$ be a function such that $\pi(i) := |\{j \mid \psi(j) = \psi(i), j \le i\}|$. Let $d$ be the maximum number of multi-use of labels in $f$, i.e., $d := \max_{i\in[n]} \pi(i)$.

$$\mathbf{r}, \mathbf{r}_1, \dots, \mathbf{r}_d \leftarrow \mathbb{Z}_p^k,\ \ [\mathbf{w}_1]_1, \dots, [\mathbf{w}_n]_1 \leftarrow \mathsf{Share}(f, [\mathbf{WBr}]_1) \in \mathbb{Z}_p^{k+1},$$
$$c_1 := [\mathbf{Br}]_2,\ \ c_{2,j} := [\mathbf{Br}_j]_2 \ \text{ for } j \in [d],\ \ c_4 := [\mathbf{r}^\top \mathbf{B}^\top \mathbf{k}]_T M,$$
$$([\mathbf{U}_{\psi(i),0}]_1, [\mathbf{U}_{\psi(i),1}]_1) := H(\psi(i)),$$
$$c_{3,i} := [\mathbf{w}_i + (x_i \mathbf{U}_{\psi(i),0} + \mathbf{U}_{\psi(i),1})\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1,$$
$$c_{3,i} := (c_{3,i,1}, c_{3,i,2}) := ([-\mathbf{w}_i + \mathbf{U}_{\psi(i),0}\mathbf{r}_{\pi(i)}]_1, [x_i \mathbf{w}_i + \mathbf{U}_{\psi(i),1}\mathbf{r}_{\pi(i)}]_1) \ \text{ if } t(i) = 0$$
$$\text{for } i \in [n],$$
$$\mathsf{ct}_x := (x, c_1, \{c_{2,j}\}_{j\in[d]}, \{c_{3,i}\}_{i\in[n]}, c_4).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$**:** It takes pk, msk, and a predicate $y = (\mathbf{y} \in \mathbb{Z}_p^m, \phi)$ and outputs $\mathsf{sk}_y$ as follows.

$$\mathbf{s} \leftarrow \mathbb{Z}_p^k,\ \ ([\mathbf{U}_{\phi(i),0}]_1, [\mathbf{U}_{\phi(i),1}]_1) := H(\phi(i)),\ \ (\mathbf{V}_{\phi(i),0}, \mathbf{V}_{\phi(i),1}) := F_K(\phi(i))$$
$$k_1 := [\mathbf{As}]_2,\ \ k_2 := [\mathbf{k} + \mathbf{W}^\top \mathbf{As}]_1,\ \ k_{3,i} := [\mathbf{B}^*(y_i \mathbf{U}^\top_{\phi(i),0} + \mathbf{U}^\top_{\phi(i),1})\mathbf{As} + \mathbf{B}^*_{12}(y_i \mathbf{V}^\top_{\phi(i),0} + \mathbf{V}^\top_{\phi(i),1})\mathbf{As}]_1 \ \text{ for } i \in [m],$$
$$\mathsf{sk}_y := (y, k_1, k_2, \{k_{3,i}\}_{i\in[m]}).$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)$**:** It takes pk, $\mathsf{ct}_x$, and $\mathsf{sk}_y$. It computes $b \in \{0,1\}^n$ from $x$ and $y$ as in Definition 2.7. If $f(b) = 0$, it outputs $\perp$. Otherwise, computes a set $S \subseteq \{i \mid b_i = 1\}$ such that $\mathbf{WBr} = \sum_{i\in S} \mathbf{w}_i$. Let $S_1 := S \cap \{i \mid t(i) = 1\}$ and $S_0 := S \cap \{i \mid t(i) = 0\}$. Then outputs $M'$ as follows.

$$D_{1,j} := e\left( \sum_{\substack{\pi(i)=j \\ i\in S_1}} c_{3,i} + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \frac{1}{x_i - y_{\phi^{-1}(\psi(i))}} (y_{\phi^{-1}(\psi(i))} c_{3,i,1} + c_{3,i,2}), k_1 \right),$$
$$D_{2,j} := e\left( \sum_{\substack{\pi(i)=j \\ i\in S_1}} k_{3,\phi^{-1}(\psi(i))} + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \frac{1}{x_i - y_{\phi^{-1}(\psi(i))}} k_{3,\phi^{-1}(\psi(i))}, c_{2,j} \right)^\top \quad \text{for } j \in [d],$$
$$M' := c_4 / \left( e(k_2, c_1)^\top / \prod_{j\in[d]} (D_{1,j}/D_{2,j}) \right).$$

**Correctness:** For honestly generated $\mathsf{ct}_x$ and $\mathsf{sk}_y$ such that $R(x, y) = 1$, we have

$$D_{1,j} = \left[ \sum_{\substack{\pi(i)=j \\ i\in S_1}} \left( \mathbf{w}_i^\top \mathbf{As} + \mathbf{r}_j^\top (x_i \mathbf{U}^\top_{\psi(i),0} + \mathbf{U}^\top_{\psi(i),1})\mathbf{As} \right) + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \left( \mathbf{w}_i^\top \mathbf{As} + \frac{1}{x_i - y_{\phi^{-1}(\psi(i))}} \mathbf{r}_j^\top (y_{\phi^{-1}(\psi(i))} \mathbf{U}^\top_{\psi(i),0} + \mathbf{U}^\top_{\psi(i),1})\mathbf{As} \right) \right]_T \in G_T,$$
$$D_{2,j} = \left[ \sum_{\substack{\pi(i)=j \\ i\in S_1}} \left( \mathbf{r}_j^\top (y_{\phi^{-1}(\psi(i))} \mathbf{U}^\top_{\psi(i),0} + \mathbf{U}^\top_{\psi(i),1})\mathbf{As} \right) + \sum_{\substack{\pi(i)=j \\ i\in S_0}} \left( \frac{1}{x_i - y_{\phi^{-1}(\psi(i))}} \mathbf{r}_j^\top (y_{\phi^{-1}(\psi(i))} \mathbf{U}^\top_{\psi(i),0} + \mathbf{U}^\top_{\psi(i),1})\mathbf{As} \right) \right]_T \in G_T.$$

In the above, we use the relations $\mathbf{B}^\top \mathbf{B}^* = \mathbf{I}_k$ and $\mathbf{B}^\top \mathbf{B}^*_{12} = \mathbf{O}_{k\times 2}$. Because $x_i = y_{\phi^{-1}(\psi(i))}$ for $i \in S_1$, we have $e(k_2, c_1)^\top / \prod_{j\in[d]} (D_{1,j}/D_{2,j}) = [\mathbf{r}^\top \mathbf{B}^\top \mathbf{k} + \mathbf{r}^\top \mathbf{B}^\top \mathbf{W}^\top \mathbf{As}]_T / [(\sum_{j\in[d]} \sum_{\substack{i\in S \\ \pi(i)=j}} \mathbf{w}_i^\top)\mathbf{As}]_T = [\mathbf{r}^\top \mathbf{B}^\top \mathbf{k}]_T$. Thus, $M' = M$.

**Figure 6: Our CP-ABE Scheme.**

semi-functional ones. In the latter process, we need to apply the indistinguishability of the core 1-ABE that rely on the sub-group assumption in the ciphertext side. However, when we apply the subgroup assumption to the ciphertext side, we cannot utilize a basis of hidden space in the secret-key side and cannot simulate semi-functional keys. To circumvent the problem, we need one more hidden space as in [12, 15]. We give a bit more detailed overview in the following.

Similarly to the proof of KP-ABE, we first replace the PRF with a random function. Then, our scheme basically follows the construction on the dual system group from prime-order groups in [11]. Concretely, we can rewrite $c_{3,i}$ and $k_{3,i}$ in the challenge ciphertext and secret keys as follows:

$$\begin{aligned}
c_{3,i} &:= [\mathbf{w}_i + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1,\\
c_{3,i} &:= \begin{pmatrix}[-\mathbf{w}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1,\\ [x_i\mathbf{w}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1\end{pmatrix}, \ \text{ if } t(i) = 0\\
k_{3,i} &:= [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{A}\mathbf{s}]_1,
\end{aligned}$$

where $\mathbf{W}_{i,b} \in \mathbb{Z}_p^{(k+1)\times(k+2)}$. After that, we first change the challenge ciphertext and then secret keys gradually into a semi-functional form. The latter part is more complicated than the corresponding process in KP-ABE. The reason is that when we apply the indistinguishability of core 1-ABE for CP-ABE (Fig 7) to change each secret key into the semi-functional form, $\mathbf{b}_1^*$ is not given to the adversary. This is because the indistinguishability of core 1-ABE for CP-ABE relies on the MDDH assumption over $(\mathbf{B}||\mathbf{b}_1)$. Thus, if we define the form of semi-functional secret keys as

$$\begin{aligned}
k_1 &:= [\mathbf{A}\mathbf{s}]_2,\ k_2 := [\mathbf{k} + \boxed{\mu\mathbf{b}_1^*} + \mathbf{W}^\top\mathbf{A}\mathbf{s}]_1,\\
k_{3,i} &:= [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{A}\mathbf{s}]_1,
\end{aligned}$$

the simulator cannot generate semi-functional secret keys. To circumvent the problem, we leverage the second hidden space $\mathbf{b}_2^*$ and define the form of semi-functional secret keys as

$$\begin{aligned}
k_1 &:= [\mathbf{A}\mathbf{s}]_2,\ k_2 := [\mathbf{k} + \boxed{\mu\mathbf{b}_2^*} + \mathbf{W}^\top\mathbf{A}\mathbf{s}]_1,\\
k_{3,i} &:= [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{A}\mathbf{s}]_1.
\end{aligned}$$

To change each secret key into the semi-functional form, we need several hybrids. Finally, we argue that the challenge ciphertext statistically hide the underlying plaintext.

Proof. We consider a series of hybrids $\mathsf{H}_0$, $\mathsf{H}_1$, $\mathsf{H}_2$, $\mathsf{H}_{3,0}$, $\mathsf{H}_{3,\iota,1}$ to $\mathsf{H}_{3,\iota,3}$ for $\iota \in \{1,\dots,q_{\mathsf{sk}}\}$, where $\mathsf{H}_0$ is the real game and $\mathsf{H}_{3,q_{\mathsf{sk}},3}$ is the final game. In the following, we denote the event $\beta = \beta'$ in hybrid $\mathsf{H}$ by $\langle \mathcal{A}, \mathsf{H}\rangle_{\mathsf{win}}$, where $\beta$ is a random bit chosen by challenger $C$, and $\beta'$ is the output of $\mathcal{A}$. Note that we have

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_0\rangle_{\mathsf{win}}] - 1/2| = \mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda). \tag{8}$$

$\underline{\mathsf{H}_1}$. We define $\mathsf{H}_1$ as the same as $\mathsf{H}_0$ except replacing PRF $F_K$ in KeyGen with a random function $R : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times 2} \times \mathbb{Z}_p^{(k+1)\times 2}$. From the definition of PRFs, we have

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_0\rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_1\rangle_{\mathsf{win}}]| \le \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_1}(\lambda). \tag{9}$$

$\underline{\mathsf{H}_2}$. Next, we define $\mathsf{H}_2$. We change the behavior of random oracle $H$ and random function $R$. Consider another random oracle $H' : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times(k+2)} \times \mathbb{Z}_p^{(k+1)\times(k+2)}$ that only challenger $C$ can access. We denote the first and second elements of $H'(i)$ by $\mathbf{W}_{i,0}$ and $\mathbf{W}_{i,1}$, respectively. In $\mathsf{H}_2$, $H(i)$ outputs $([\mathbf{W}_{i,0}\mathbf{B}]_1, [\mathbf{W}_{i,1}\mathbf{B}]_1)$, and $R(i)$ outputs $(\mathbf{W}_{i,0}\mathbf{B}_{12}, \mathbf{W}_{i,1}\mathbf{B}_{12})$. Then, we have

$$\Pr[\langle \mathcal{A}, \mathsf{H}_1\rangle_{\mathsf{win}}] = \Pr[\langle \mathcal{A}, \mathsf{H}_2\rangle_{\mathsf{win}}]. \tag{10}$$

It is not difficult to confirm that the above equality holds because $\overline{\mathbf{B}} = (\mathbf{B}||\mathbf{B}_{12}) \in \mathbb{Z}_p^{(k+2)\times(k+2)}$ is a regular matrix, and thus $\mathbf{W}_{i,b}\overline{\mathbf{B}}$ is randomly distributed in $\mathbb{Z}_p^{(k+1)\times(k+2)}$ for $\mathcal{A}$. By this conceptual change, we can rewrite $c_{3,i}$ and $k_{3,i}$ in the challenge ciphertext and secret keys as follows;

$$\begin{aligned}
c_{3,i} &:= [\mathbf{w}_i + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1,\\
c_{3,i} &:= \begin{pmatrix}[-\mathbf{w}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1,\\ [x_i\mathbf{w}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1\end{pmatrix}, \ \text{ if } t(i) = 0\\
k_{3,i} &:= [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{A}\mathbf{s}]_1.
\end{aligned}$$

In the above, we use the relations $\mathbf{B}^*\mathbf{B}^\top + \mathbf{B}^*_{12}\mathbf{B}^\top_{12} = \mathbf{I}_{k+2}$.

$\underline{\mathsf{H}_{3,\iota}}$. To describe $\mathsf{H}_{3,0}$ and $\mathsf{H}_{3,\iota,1}$ to $\mathsf{H}_{3,\iota,3}$, we define some distributions on ciphertexts and secret keys as follows. Concretely, we define two types of ciphertexts and four types of secret keys. For ciphertexts, we define a normal ciphertext and semi-functional (SF) ciphertext. A normal ciphertext is one generated as in $\mathsf{H}_2$. That is,

$$\begin{aligned}
&c_1 = [\mathbf{B}\mathbf{r}]_2,\ \ c_{2,j} = [\mathbf{B}\mathbf{r}_j]_2,\\
&[\mathbf{w}_1]_1,\dots,[\mathbf{w}_n]_1 \leftarrow \mathsf{Share}(f, [\mathbf{W}\mathbf{B}\mathbf{r}]_1),\\
&c_{3,i} := [\mathbf{w}_i + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1,\\
&c_{3,i} := \begin{pmatrix}[-\mathbf{w}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1,\\ [x_i\mathbf{w}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1\end{pmatrix}, \ \text{ if } t(i) = 0\\
&c_4 = [\mathbf{r}^\top\mathbf{B}^\top\mathbf{k}]_T M.
\end{aligned}$$

An SF ciphertext is the same as the normal one except that $\mathbf{B}\mathbf{r}$ is replaced with $\mathbf{d} \leftarrow \mathbb{Z}_p^{k+2}$. That is,

$$\begin{aligned}
&c_1 = [\boxed{\mathbf{d}}]_2,\ \ c_{2,j} = [\mathbf{B}\mathbf{r}_j]_2,\\
&[\mathbf{w}_1]_1,\dots,[\mathbf{w}_n]_1 \leftarrow \mathsf{Share}(f, [\mathbf{W}\boxed{\mathbf{d}}]_1),\\
&c_{3,i} := [\mathbf{w}_i + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1,\\
&c_{3,i} := \begin{pmatrix}[-\mathbf{w}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1,\\ [x_i\mathbf{w}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1\end{pmatrix}, \ \text{ if } t(i) = 0\\
&c_4 = [\boxed{\mathbf{d}^\top}\mathbf{k}]_T M.
\end{aligned} \tag{11}$$

For secret keys, we define four secret keys, namely, normal, P-normal, P-SF, and SF. That is, $\mathsf{sk}_y$ is defined as

$$\begin{aligned}
\text{normal: } &\begin{pmatrix} k_1 := [\mathbf{As}]_2,\ k_2 := [\mathbf{k} + \mathbf{W}^\top \mathbf{As}]_1, \\ k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1 \end{pmatrix}, \\
\text{P-normal: } &\begin{pmatrix} k_1 := [\boxed{\mathbf{c}}]_2,\ k_2 := [\mathbf{k} + \mathbf{W}^\top \boxed{\mathbf{c}}]_1, \\ k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\boxed{\mathbf{c}}]_1 \end{pmatrix}, \\
\text{P-SF: } &\begin{pmatrix} k_1 := [\mathbf{c}]_2,\ k_2 := [\mathbf{k} + \boxed{\mu \mathbf{b}^*_2} + \mathbf{W}^\top \mathbf{c}]_1, \\ k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{c}]_1 \end{pmatrix}, \\
\text{SF: } &\begin{pmatrix} k_1 := [\boxed{\mathbf{As}}]_2,\ k_2 := [\mathbf{k} + \mu \mathbf{b}^*_2 + \mathbf{W}^\top \boxed{\mathbf{As}}]_1, \\ k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\boxed{\mathbf{As}}]_1 \end{pmatrix},
\end{aligned} \tag{12}$$

where $\mu \leftarrow \mathbb{Z}_p$ and $\mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$. Then, we define $\mathsf{H}_{3,0}$ and $\mathsf{H}_{3,\iota,1}$ to $\mathsf{H}_{3,\iota,3}$ for $\iota \in \{1, \ldots, q_{\mathsf{sk}}\}$ as follows.

$\mathsf{H}_{3,0}$: The challenge ciphertext is SF, and all secret keys are normal.

$\mathsf{H}_{3,\iota,1}$: The challenge ciphertext is SF, the first $\iota - 1$ secret keys are SF, the $\iota$-th secret key is P-normal.

$\mathsf{H}_{3,\iota,2}$: The challenge ciphertext is SF, the first $\iota - 1$ secret keys are SF, and the $\iota$-th secret key is P-SF.

$\mathsf{H}_{3,\iota,3}$: The challenge ciphertext is SF, the first $\iota - 1$ secret keys are SF, and the $\iota$-th secret key is SF.

Lemma 5.3.

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_2 \rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_{3,0} \rangle_{\mathsf{win}}]| \leq \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda). \tag{13}$$

Proof. The difference between these hybrids is whether the challenge ciphertext is normal or SF. To prove the lemma, we describe $\mathcal{B}$, which is given an instance of the $\mathcal{U}_{k+2,k}$-MDDH problem $(\mathbb{G}, [\mathbf{B}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$.

(1) $\mathcal{B}$ generates $\mathbf{A}$, $\mathbf{W}$, and $\mathbf{k}$ by itself.
(2) $\mathcal{B}$ computes $\mathsf{pk} = (\mathbb{G}, [\mathbf{B}]_2, [\mathbf{WB}]_1, e([\mathbf{B}]_1, [\mathbf{k}]_2))$ and gives it to $\mathcal{A}$.
(3) For a query $H(i)$, $\mathcal{B}$ answers with $([\mathbf{W}_{i,0}\mathbf{B}]_1, [\mathbf{W}_{i,1}\mathbf{B}]_1)$, where $(\mathbf{W}_{i,0}, \mathbf{W}_{i,1})$ is an output of $H'(i)$.
(4) For a query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes $\mathsf{sk}_y$ as a normal one in Eq.(12). Note that $\mathcal{B}$ can generate sk without the random function $R$ because it does not contain terms related to $\mathbf{B}$ any more.
(5) For the challenge query with the attribute $x^* = (\mathbf{x}, f, \psi, t)$, $\mathcal{B}$ flip the coin $\delta \leftarrow \{0,1\}$ and generates $\mathsf{ct}_{x^*}$ as

$$\begin{aligned}
&c_1 = [\mathbf{t}_\beta]_2,\ \ c_{2,j} = [\mathbf{B}\mathbf{r}_j]_2, \\
&[\mathbf{w}_1]_1, \ldots, [\mathbf{w}_n]_1 \leftarrow \mathsf{Share}(f, [\mathbf{W}\mathbf{t}_\beta]_1), \\
&c_{3,i} := [\mathbf{w}_i + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1, \\
&c_{3,i} := \begin{pmatrix} [-\mathbf{w}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [x_i \mathbf{w}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix}, \ \text{ if } t(i) = 0 \\
&c_4 = e([\mathbf{t}_\beta]_1, [\mathbf{k}]_2) M_\delta.
\end{aligned}$$

(6) $\mathcal{B}$ outputs $\mathsf{true}(\delta = \delta')$, where $\delta'$ is an output of $\mathcal{A}$.

Clearly, the case $\beta = 0$ corresponds to $\mathsf{H}_2$ and the case $\beta = 1$ corresponds to $\mathsf{H}_{3,0}$. Because $\mathsf{Adv}^{\mathcal{U}_{k+2,k}\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda) \leq \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda)$, the lemma holds. □

Lemma 5.4. *Let $\mathsf{H}_{3,0} = \mathsf{H}_{3,0,3}$. For $\iota \in [q_{\mathsf{sk}}]$, we have*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota-1,3} \rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota,1} \rangle_{\mathsf{win}}]| \leq \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda). \tag{14}$$

Proof. The difference between these hybrids is whether the $\iota$-th secret key is normal or P-normal. We describe $\mathcal{B}$, which is given an instance of $\mathcal{D}_k$-MDDH problem, $(\mathbb{G}, [\mathbf{A}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$.

(1) $\mathcal{B}$ generates $\mathbf{B}$, $\mathbf{b}^*_2$ $\mathbf{W}$, and $\mathbf{k}$ by itself.
(2) $\mathcal{B}$ computes $\mathsf{pk} = (\mathbb{G}, [\mathbf{B}]_2, [\mathbf{WB}]_1, e([\mathbf{B}]_1, [\mathbf{k}]_2))$ and gives it to $\mathcal{A}$.
(3) For a query $H(i)$, $\mathcal{B}$ answers with $([\mathbf{W}_{i,0}\mathbf{B}]_1, [\mathbf{W}_{i,1}\mathbf{B}]_1)$, where $(\mathbf{W}_{i,0}, \mathbf{W}_{i,1})$ is an output of $H'(i)$.
(4) For the $\tau$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$ such that $\tau < \iota$, $\mathcal{B}$ computes $\mathsf{sk}_y$ as

$$\begin{aligned}
&k_1 := [\mathbf{As}]_2,\ k_2 := [\mathbf{k} + \mu \mathbf{b}^*_2 + \mathbf{W}^\top \mathbf{As}]_1, \\
&k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1,
\end{aligned}$$

where $\mu \leftarrow \mathbb{Z}_p$.
(5) For the $\iota$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes $\mathsf{sk}_y$ as

$$\begin{aligned}
&k_1 := [\mathbf{t}_\beta]_2,\ k_2 := [\mathbf{k} + \mathbf{W}^\top \mathbf{t}_\beta]_1, \\
&k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{t}_\beta]_1.
\end{aligned}$$

(6) For the $\tau$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$ such that $\tau > \iota$, $\mathcal{B}$ computes $\mathsf{sk}_y$ as

$$\begin{aligned}
&k_1 := [\mathbf{As}]_2,\ k_2 := [\mathbf{k} + \mathbf{W}^\top \mathbf{As}]_1, \\
&k_{3,i} := [(y_i \mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1.
\end{aligned}$$

(7) For the challenge query, $\mathcal{B}$ flip the coin $\delta \leftarrow \{0,1\}$ and generates $\mathsf{ct}_{x^*}$ for $M_\delta$ as in Eq.(11).
(8) $\mathcal{B}$ outputs $\mathsf{true}(\delta = \delta')$, where $\delta'$ is an output of $\mathcal{A}$.

Clearly, the case $\beta = 0$ corresponds to $\mathsf{H}_{3,\iota-1,3}$ and the case $\beta = 1$ corresponds to $\mathsf{H}_{3,\iota,1}$. □

Before we move to the next lemma, we define a variant of the core 1-ABE game as in Fig 7. The variant $\mathrm{G}^{1\text{-ABE+}}_\beta$ is different from the original one essentially in the part framed by dashed boxes, except the sizes of matrices. The conditions for queries are the same as in Definition 3.2. Then, Lemma 5.5 holds. We present the proof of Lemma 5.5 in Section 5.3, but it is essentially the same as the proof of Lemma 3.3.

Lemma 5.5. *Let $B$ be the maximum depth of formula $f$ for all choice of $f$ by $\mathcal{A}$. For any PPT adversary $\mathcal{A}$, there exists a PPT algorithm $\mathcal{B}$ such that*

$$\begin{aligned}
\mathsf{Adv}^{1\text{-ABE+}}_{\mathcal{A}}(\lambda) &:= |\Pr[\langle \mathcal{A}, \mathrm{G}^{1\text{-ABE+}}_0 \rangle = 1] - \Pr[\langle \mathcal{A}, \mathrm{G}^{1\text{-ABE+}}_1 \rangle = 1]| \\
&\leq 2^{9B+2}(\mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda) + 2^{-\Omega(\lambda)}).
\end{aligned}$$

Lemma 5.6. *For $\iota \in [q_{\mathsf{sk}}]$, we have*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota,1} \rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota,2} \rangle_{\mathsf{win}}]| \leq \mathsf{Adv}^{1\text{-ABE+}}_{\mathcal{B}}(\lambda). \tag{15}$$

Proof. In contrast to the proof of our KP-ABE, $\mathcal{B}$ uses $O_X$ to generate the $\iota$-th secret key and $O_F$ to generate the challenge ciphertext. Thus, the reduction can proves indistinguishability of two types of ciphertexts. We show that this indistinguishability

$\underline{\mathsf{G}^{\text{1-ABE+}}_{\beta}}$

$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mu' \leftarrow \mathbb{Z}_p$, $\mathbf{A} \leftarrow \mathcal{D}_k$, $\overline{\mathbf{B}} \leftarrow \mathsf{GL}_{k+2}(\mathbb{Z}_p)$,
$\boxed{\mathbf{d} \leftarrow \mathbb{Z}_p^{k+2},\ \ \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}}$, $L := \emptyset$,
$\beta' \leftarrow \mathcal{A}^{O_X(\cdot), O_F(\cdot), O_R(\cdot)}(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \boxed{\mathbf{b}_2^*, \mathbf{d}, \mathbf{W}})$

$\underline{O_X(\cdot)}$
Input: $y = (\mathbf{y} \in \mathbb{Z}_p^m, \phi) \in \mathcal{Y}$
$A_0 := \mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$
For $i \in [m]$:
  If $(\phi(i), *, *) \notin L$:
    $\mathbf{W}_{\phi(i),0}, \mathbf{W}_{\phi(i),1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}$
    $L := L \cup (\phi(i), \mathbf{W}_{\phi(i),0}, \mathbf{W}_{\phi(i),1})$
  $A_i := (y_i \mathbf{W}_{\phi(i),0}^\top + \mathbf{W}_{\phi(i),1}^\top)\mathbf{c}$
Output $(A_0, \{A_i\}_{i\in[m]})$

$\underline{O_F(\cdot)}$
Input: $x = (\mathbf{x} \in \mathbb{Z}_p^n, f, \psi, t) \in \mathcal{X}$
$\mathbf{w}_1, \dots, \mathbf{w}_n \leftarrow \mathsf{Share}(f, \boxed{\mathbf{W}\mathbf{d}})$
$\sigma_1, \dots, \sigma_n \leftarrow \boxed{\mathsf{Share}(f, \beta\mu')}$
$\pi(i) := |\{j | \psi(j) = \psi(i), j \le i\}|$
$d := \max_{i\in[n]} \pi(i)$
$\mathbf{r}_1, \dots, \mathbf{r}_d \leftarrow \mathbb{Z}_p^k$
$P_0 := ([\mathbf{B}\mathbf{r}_1]_2, \dots, [\mathbf{B}\mathbf{r}_d]_2)$
For $i \in [n]$:
  If $(\psi(i), *, *) \notin L$:
    $\mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}$
    $L := L \cup (\psi(i), \mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1})$
  If $t(i) = 1$ :
    $P_i := [\mathbf{w}_i + \sigma_i \mathbf{a}^\perp + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1$
  Else:
    $P_i := \begin{pmatrix} [-(\mathbf{w}_i + \sigma_i \mathbf{a}^\perp) + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [x_i(\mathbf{w}_i + \sigma_i \mathbf{a}^\perp) + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix}$
Output $(P_0, \{P_i\}_{i\in[n]})$

$\underline{O_R(\cdot)}$
Input: $i \in \{0,1\}^*$
If $(i, *, *) \notin L$:
  $\mathbf{W}_{i,0}, \mathbf{W}_{i,1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}$, $L := L \cup (i, \mathbf{W}_{i,0}, \mathbf{W}_{i,1})$
Output $\boxed{([\mathbf{W}_{i,0}\mathbf{B}]_1, [\mathbf{W}_{i,1}\mathbf{B}]_1)}$

**Figure 7: Core 1-ABE game for our CP-ABE scheme.**

is equivalent to that between the cases where the $\iota$-th secret key is P-normal and P-SF1. We describe $\mathcal{B}$ in the following.

(1) $\mathcal{B}$ is given $(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{b}_2^*, \mathbf{d}, \mathbf{W})$ from the 1-ABE game.
(2) $\mathcal{B}$ generates $\mathbf{k} \leftarrow \mathbb{Z}_p^{k+2}$ and gives $\mathsf{pk} = (\mathbb{G}, [\mathbf{B}]_2, [\mathbf{W}\mathbf{B}]_1, e([\mathbf{B}]_1, [\mathbf{k}]_2))$ to $\mathcal{A}$.
(3) For a query $H(i)$, $\mathcal{B}$ answers with the reply of $O_R(i)$.
(4) For the challenge query with an attribute $x^*$, $\mathcal{B}$ flip the coin $\delta \leftarrow \{0,1\}$. Then, $\mathcal{B}$ obtains $(P_0, \{P_i\}_{i\in[n]})$ as the reply of $O_F(x^*)$. $\mathcal{B}$ returns $\mathsf{ct}_{x^*}$ computing as

$$\mathsf{ct}_{x^*} := \left([\mathbf{d}]_2, P_0, \{P_i\}_{i\in[n]}, [\mathbf{d}^\top \mathbf{k}]_T M_\delta\right).$$

(5) For the $\tau$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\tau < i$ and $y = (\mathbf{y}, \phi)$, $\mathcal{B}$ compute $\mathsf{sk}_y$ as an SF secret key in Eq. (12) with a fresh randomness $\mu \leftarrow \mathbb{Z}_p$.
(6) For the $\tau$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\tau = i$ and $y = (\mathbf{y}, \phi)$, $\mathcal{B}$ obtains $(P_0, \{P_i\}_{i\in[n]})$ as the reply of $O_X(y)$. Then, $\mathcal{B}$ returns $\mathsf{sk}_y$ computing as

$$\mathsf{sk}_y := ([A_0]_2, [\mathbf{k} + \mathbf{W}^\top A_0]_1, \{[A_i]_1\}_{i\in[m]}).$$

(7) For the $\tau$-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\tau > i$ and $y = (\mathbf{y}, \phi)$, $\mathcal{B}$ compute $\mathsf{sk}_y$ as a normal secret key in Eq.(12).
(8) $\mathcal{B}$ outputs $\mathsf{true}(\delta = \delta')$, where $\delta'$ is an output of $\mathcal{A}$.

Then, we implicitly define that $\mathbf{W} := \widetilde{\mathbf{W}} - \frac{\beta\mu' \mathbf{a}^\perp \mathbf{b}_2^{*\top}}{\mathbf{b}_2^{*\top}\mathbf{d}}$ where $\widetilde{\mathbf{W}} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}$. Note that the new definition does not change the distribution of $\mathbf{W}$. By the definition, the distributions that $\mathcal{A}$ obtains can be written as

$$\mathsf{pk} = (\mathbb{G}, [\mathbf{B}]_2, [\widetilde{\mathbf{W}}\mathbf{B}]_1, e([\mathbf{B}]_1, [\mathbf{k}]_2)),$$

$$\mathsf{sk}_y = \begin{cases} \begin{pmatrix} k_1 := [\mathbf{A}\mathbf{s}]_2,\ k_2 := [\mathbf{k} + \mu\mathbf{b}_2^* + \widetilde{\mathbf{W}}^\top \mathbf{A}\mathbf{s}]_1, \\ k_{3,i} := [(y_i \mathbf{W}_{\phi(i),0}^\top + \mathbf{W}_{\phi(i),1}^\top)\mathbf{A}\mathbf{s}]_1 \end{pmatrix} & \tau < \iota \\ \begin{pmatrix} k_1 := [\mathbf{c}]_2,\ k_2 := [\mathbf{k} \underbrace{- \frac{\beta\mu' \mathbf{b}_2^* \mathbf{a}^{\perp\top} \mathbf{c}}{\mathbf{b}_2^{*\top}\mathbf{d}}}_{:=\beta\mu\mathbf{b}_2^*} + \widetilde{\mathbf{W}}^\top \mathbf{c}]_1, \\ k_{3,i} := [(y_i \mathbf{W}_{\phi(i),0}^\top + \mathbf{W}_{\phi(i),1}^\top)\mathbf{c}]_1 \end{pmatrix} & \tau = \iota \\ \begin{pmatrix} k_1 := [\mathbf{A}\mathbf{s}]_2,\ k_2 := [\mathbf{k} + \widetilde{\mathbf{W}}^\top \mathbf{A}\mathbf{s}]_1, \\ k_{3,i} := [(y_i \mathbf{W}_{\phi(i),0}^\top + \mathbf{W}_{\phi(i),1}^\top)\mathbf{A}\mathbf{s}]_1 \end{pmatrix} & \tau > \iota \end{cases}.$$

Next, we look at the distribution of $\mathsf{ct}_{x^*}$. From Lemma 3.1, we have

$$\mathsf{ct}_{x^*} = \begin{pmatrix} c_1 = [\mathbf{d}]_2,\ \ c_{2,j} = [\mathbf{B}\mathbf{r}_j]_2, \\ c_{3,i} := [\mathbf{w}_i + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \ \text{ if } t(i) = 1, \\ c_{3,i} := \begin{pmatrix} [-\mathbf{w}_i + \mathbf{W}_{\psi(i),0}\mathbf{B}\mathbf{r}_{\pi(i)}]_1, \\ [x_i \mathbf{w}_i + \mathbf{W}_{\psi(i),1}\mathbf{B}\mathbf{r}_{\pi(i)}]_1 \end{pmatrix}, \ \text{ if } t(i) = 0 \\ c_4 = [\mathbf{d}^\top \mathbf{k}]_T M_\delta. \end{pmatrix},$$

where $\mathbf{w}_1, \dots, \mathbf{w}_n \leftarrow \mathsf{Share}(f, \mathbf{W}\mathbf{d} + \beta\mu'\mathbf{a}^\perp) = \mathsf{Share}(f, \widetilde{\mathbf{W}}\mathbf{d})$. In the above, we use the relations $\mathbf{a}^{\perp\top}\mathbf{A} = \mathbf{0}^\top$ and $\mathbf{b}_2^{*\top}\mathbf{B} = \mathbf{0}^\top$.

Observe that $\mathcal{A}$'s view corresponds to $\mathsf{H}_{3,\iota,1}$ if $\beta = 0$ and it corresponds to $\mathsf{H}_{3,\iota,2}$ otherwise, by setting $\mu := -\frac{\mu' \mathbf{a}^{\perp\top} \mathbf{c}}{\mathbf{b}_2^{*\top}\mathbf{d}}$. Note that $\mu'$ appear only in $k_2$ in the $\iota$-th secret key. Thus, $\mu$ is randomly distributed in $\mathbb{Z}_p$. This concludes the proof. □

Lemma 5.7. *For $\iota \in [q_{\mathsf{sk}}]$, we have*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota,2} \rangle_{\mathsf{win}}] - \Pr[\langle \mathcal{A}, \mathsf{H}_{3,\iota,3} \rangle_{\mathsf{win}}]| \le \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda). \tag{16}$$

We omit the proof because this lemma can be proven similarly to Lemma 5.4.

Lemma 5.8.

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{3,q_{\mathsf{sk}},3} \rangle_{\mathsf{win}}] - 1/2| \le 2^{-\Omega(\lambda)}. \tag{17}$$

$\widehat{\mathsf{H}}_1^{h_{\iota-1}}$, $\boxed{\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}}$, $\boxed{\widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}}}$, $\boxed{\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}}$

$\underline{O_F(\cdot)}$

Input: $x = (\mathbf{x} \in \mathbb{Z}_p^n, f, \psi, t) \in \mathcal{X}$

$\mathbf{w}_1, \dots, \mathbf{w}_n \leftarrow \mathsf{Share}(f, \mathbf{Wd})$, $\sigma_1, \dots, \sigma_n \leftarrow \widetilde{\mathsf{Share}}(f, \mu, u_0)$

$\pi(i) := |\{j \mid \psi(j) = \psi(i), j \le i\}|$

$d := \max_{i \in [n]} \pi(i)$

$\mathbf{r}_1, \dots, \mathbf{r}_d \leftarrow \mathbb{Z}_p^k$, $\mathbf{f} \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1)$

$\mathbf{v}_i := \mathbf{B}\mathbf{r}_i$ for $i \in [d]$

$\boxed{\mathbf{v}_i := \mathbf{B}\mathbf{r}_i \text{ for } i \in [d] \backslash \pi(i^*),\ \mathbf{v}_{\pi(i^*)} := \mathbf{f}}$

$P_0 := ([\mathbf{v}_1]_2, \dots, [\mathbf{v}_d]_2)$

For $i \in [n]$:

  If $(\psi(i), *, *) \notin L$:

    $\mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}$

    $L := L \cup (\psi(i), \mathbf{W}_{\psi(i),0}, \mathbf{W}_{\psi(i),1})$

  If $i = i^*$

    $\boxed{u \leftarrow \mathbb{Z}_p,\ \sigma_i := \sigma_i + u}$

  If $t(i) = 1$ :

    $P_i := [\mathbf{w}_i + \sigma_i \mathbf{a}^\perp + (x_i \mathbf{W}_{\psi(i),0} + \mathbf{W}_{\psi(i),1})\mathbf{v}_{\pi(i)}]_1$

  Else:

    $P_i := \begin{pmatrix} [-(\mathbf{w}_i + \sigma_i \mathbf{a}^\perp) + \mathbf{W}_{\psi(i),0}\mathbf{v}_{\pi(i)}]_1, \\ [x_i(\mathbf{w}_i + \sigma_i \mathbf{a}^\perp) + \mathbf{W}_{\psi(i),1}\mathbf{v}_{\pi(i)}]_1 \end{pmatrix}$

Output $(P_0, \{P_i\}_{i \in [n]})$

**Figure 8: Description of $O_F$ in hybrids.**

Proof. Because $(\mathbf{B}^* || \mathbf{b}_1^* || \mathbf{b}_2^*)$ forms a basis of $\mathbb{Z}_p^{k+2}$, redefining $\mathbf{k}$ as $\mathbf{k} := \mathbf{B}^*\mathbf{z} + z_1\mathbf{b}_1^* + z_2\mathbf{b}_2^*$ where $\mathbf{z} \leftarrow \mathbb{Z}_p^k$, $z_1, z_2 \leftarrow \mathbb{Z}_p$ does not change its distribution. Recall that the information on $\mathbf{k}$ that $\mathcal{A}$ obtains throughout the game is $\mathbf{B}^\top\mathbf{k}$ in $\mathsf{pk}$, $\mathbf{k} + \mu\mathbf{b}_2^*$ in $\mathsf{sk}_y$, and $\mathbf{d}^\top\mathbf{k}$ in $\mathsf{ct}_{x^*}$. However, $\mathbf{B}^\top\mathbf{k}$ does not contain the information on $z_2$ because $\mathbf{B}^\top\mathbf{b}_2^* = \mathbf{0}$. Similarly, each $\mathbf{k} + \mu\mathbf{b}_2^*$ also does not contain information on $z_2$ because it is masked by the fresh randomness $\mu$. Thus, $z_2\mathbf{d}^\top\mathbf{b}_2^*$ is randomly distributed in $\mathbb{Z}_p$ for $\mathcal{A}$, and so is $\mathbf{d}^\top\mathbf{k}$, unless $\mathbf{d}^\top\mathbf{b}_2^* = 0$. Since $\mathbf{d}$ is randomly chosen from $\mathbb{Z}_p^{k+2}$, $\mathbf{d}^\top\mathbf{b}_2^* = 0$ with a probability $2^{-\Omega(\lambda)}$. If it is not the case, $\mathsf{ct}_{x^*}$ does not have information on $\beta$, and the lemma holds. □

Thanks to Eq. (8) to (10) and (13) to (17) and Lemma 5.5, Theorem 5.2 holds. □

## 5.3 Proof of Lemma 5.5

Proof. Definitions of all hybrids and the process of the proof are the same as in the proof of Lemma 3.3. The difference lies in the part to show $\widehat{\mathsf{H}}_1^{h_{\iota-1}} \approx_c \widehat{\mathsf{H}}_0^{h_\iota}$ for $\iota \in [L]$. As well as the original proof, we define $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ to $\widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}}$, which are different from $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ only in the procedure in $O_F$ as shown in Fig 8. Then, we show indistinguishability of each pair of hybrids.

Lemma 5.9.

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_1^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}} \rangle = 1]| \le \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda).$$

Proof. The difference between these hybrids is that $\mathbf{v}_{\pi(i^*)} := \mathbf{B}\mathbf{r}_{\pi(i^*)}$ for $\mathbf{r}_{\pi(i^*)} \leftarrow \mathbb{Z}_p^k$ in the former and $\mathbf{v}_{\pi(i^*)} := \mathbf{f}$ for $\mathbf{f} \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1)$ in the latter. We show that the $\mathcal{D}_k$-MDDH problem is reduced to this difference. The reduction algorithm $\mathcal{B}$ is given an instance $(\mathbb{G}, [\mathbf{M}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$ where $\mathbf{t}_0 = \mathbf{M}\mathbf{u}$ and $\mathbf{t}_1 = \mathbf{v}$, where $\mathbf{u} \leftarrow \mathbb{Z}_p^k$ and $\mathbf{v} \leftarrow \mathbb{Z}_p^{k+1}$. Then, $\mathcal{B}$ chooses $\mathbf{X} \leftarrow \mathsf{GL}_{k+2}(\mathbb{Z}_p)$ and sets

$$\overline{\mathbf{B}} := \mathbf{X}\begin{pmatrix} \overline{\mathbf{M}} & & \\ \underline{\mathbf{M}} & 1 & \\ & & 1 \end{pmatrix},$$

$$(\overline{\mathbf{B}}^\top)^{-1} := (\mathbf{X}^\top)^{-1}\begin{pmatrix} (\overline{\mathbf{M}}^\top)^{-1} & -(\overline{\mathbf{M}}^\top)^{-1}\underline{\mathbf{M}}^\top & \\ & 1 & \\ & & 1 \end{pmatrix},$$

where $\overline{\mathbf{M}}$ is the matrix consists of the first $k$ rows of $\mathbf{M}$, and $\underline{\mathbf{M}}$ is that consists of the last row of $\mathbf{M}$. Then, $\mathcal{B}$ can computes

$$[\mathbf{B}]_{1,2} = \left[\mathbf{X}\begin{pmatrix}\mathbf{M}\\ \mathbf{0}^\top\end{pmatrix}\right]_{1,2},\quad \mathbf{b}_2^* = (\mathbf{X}^\top)^{-1}\begin{pmatrix}\mathbf{0}\\ 1\end{pmatrix}.$$

$\mathcal{B}$ generate $\mathbf{A}$, $\mathbf{d}$, and $\mathbf{W}$ by itself and gives $(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{b}_2^*, \mathbf{d}, \mathbf{W})$ to $\mathcal{A}$ as its input. When $\mathcal{A}$ queries $O_X$ and $O_R$, $\mathcal{B}$ replies honestly. When $\mathcal{A}$ queries $O_F$, $\mathcal{B}$ replies honestly except that it sets

$$[\mathbf{v}_{\pi(i^*)}]_{1,2} := \left[\mathbf{X}\begin{pmatrix}\mathbf{t}_\beta\\ 0\end{pmatrix}\right]_{1,2}$$

Because we can write

$$\mathbf{t}_\beta = \begin{pmatrix}\overline{\mathbf{M}}\\ \underline{\mathbf{M}}\end{pmatrix}\mathbf{u} + \beta u\begin{pmatrix}\mathbf{0}\\ 1\end{pmatrix},$$

where $\mathbf{u} \leftarrow \mathbb{Z}_p^k$ and $u \leftarrow \mathbb{Z}_p$, $\mathbf{v}_{\pi(i^*)}$ is uniformly distributed in $\mathsf{span}(\mathbf{B})$ if $\beta = 0$, and in $\mathsf{span}(\mathbf{B}, \mathbf{b}_1)$ otherwise. Thus, the view of $\mathcal{A}$ corresponds to $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ if $\beta = 0$, and $\widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}}$ otherwise. This concludes the proof. □

Lemma 5.10.

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,1}^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}} \rangle = 1] \le 2^{-\Omega(\lambda)}.$$

Proof. The difference between these hybrids is that a random value $u$ is added to the share $\sigma_{i^*}$ in $\widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}}$. Similarly to the previous lemma, these hybrids are identical when $O$ is not an input wire. We redefine that $\mathbf{W}_{\psi(i^*),b} := \widetilde{\mathbf{W}}_{\psi(i^*),b} + w_{\psi(i^*),b}\mathbf{a}^\perp\mathbf{b}_1^{*\top}$, where $\widetilde{\mathbf{W}}_{\psi(i^*),b} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+2)}$, $w_{\psi(i^*),b} \leftarrow \mathbb{Z}_p$, and $b \in \{0,1\}$. Since $\widetilde{\mathbf{W}}_{\psi(i^*),b}$ is chosen randomly, the distribution of redefined $\mathbf{W}_{\psi(i^*),b}$ is identical to that of the original definition. Observe that this change does not affect the outputs of $O_R$ because $\mathbf{b}_1^{*\top}\mathbf{B} = \mathbf{0}^\top$. For $O_F$, $P_i$ for $i \in \psi^{-1}(\psi(i^*))$ can be written as

If $t(i) = 1$ :

$$P_i := \begin{bmatrix} \mathbf{w}_i + \sigma_i\mathbf{a}^\perp + (y_i\widetilde{\mathbf{W}}_{\psi(i^*),0} + \widetilde{\mathbf{W}}_{\psi(i^*),1})\mathbf{v}_{\pi(i)} \\ +(x_i w_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{a}^\perp\mathbf{b}_1^{*\top}\mathbf{v}_{\pi(i)} \end{bmatrix}_1$$

Else:

$$P_i := \begin{pmatrix} [-(\mathbf{w}_i + \sigma_i\mathbf{a}^\perp) + \widetilde{\mathbf{W}}_{\psi(i^*),0}\mathbf{v}_{\pi(i)} + w_{\psi(i^*),0}\mathbf{a}^\perp\mathbf{b}_1^{*\top}\mathbf{v}_{\pi(i)}]_1, \\ [x_i(\mathbf{w}_i + \sigma_i\mathbf{a}^\perp) + \widetilde{\mathbf{W}}_{\psi(i^*),1}\mathbf{v}_{\pi(i)} + w_{\psi(i^*),1}\mathbf{a}^\perp\mathbf{b}_1^{*\top}\mathbf{v}_{\pi(i)}]_1 \end{pmatrix}.$$

For $i \neq i^*$, we have $\mathbf{b}_1^{*\top}\mathbf{v}_{\pi(i)} = \mathbf{b}_1^{*\top}\mathbf{B}\mathbf{r}_{\pi(i)} = 0$, and thus the distribution is not changed. For $i = i^*$, we have $\mathbf{b}_1^{*\top}\mathbf{v}_{\pi(i^*)} = \mathbf{b}_1^{*\top}\mathbf{d} \neq 0$ with overwhelming probability because $\mathbf{d}$ is chosen randomly from $\mathbb{Z}_p^{k+2}$.

Then, we consider the two cases.

- $t(i^*) = 1$. This case means that $\mathcal{A}$ either does not obtain an information on $\mathbf{W}_{\psi(i^*),b}$ or obtains a vector

  $$(x\mathbf{W}_{\psi(i^*),0}^\top + \mathbf{W}_{\psi(i^*),1}^\top)\mathbf{c}$$
  $$=(x\widetilde{\mathbf{W}}_{\psi(i^*),0}^\top + \widetilde{\mathbf{W}}_{\psi(i^*),1}^\top)\mathbf{c} + (yw_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}_1^*\mathbf{a}^{\perp\top}\mathbf{c}$$

  for some $y \neq x_{i^*}$ from $O_X$. In both cases, the value $(y_i w_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}_1^{*\top}\mathbf{d}$ in $P_{i^*}$ is randomly distributed from the viewpoint of $\mathcal{A}$ because this is a pairwise independent function. Thus, adding $u\mathbf{a}^\perp$ to $P_{i^*}$ does not change the distribution.
- $t(i^*) = 0$. This case means that $\mathcal{A}$ either does not obtain an information on $\mathbf{W}_{\psi(i^*),b}$ or obtains a vector

  $$(x\mathbf{W}_{\psi(i^*),0}^\top + \mathbf{W}_{\psi(i^*),1}^\top)\mathbf{c}$$
  $$=(x\widetilde{\mathbf{W}}_{\psi(i^*),0}^\top + \widetilde{\mathbf{W}}_{\psi(i^*),1}^\top)\mathbf{c} + (yw_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}_1^*\mathbf{a}^{\perp\top}\mathbf{c}$$

  for $y = x_{i^*}$. In both cases, setting $w_{\psi(i^*),0} := w'_{\psi(i^*),0} - u/\mathbf{b}_1^{*\top}\mathbf{d}$ and $w_{\psi(i^*),1} := w'_{\psi(i^*),1} + y_i u/\mathbf{b}_1^{*\top}\mathbf{d}$ for randomly chosen $w'_{\psi(i^*),b}$ does not change the distribution.

Thus, the views of $\mathcal{A}$ in both hybrids are identical unless $\mathbf{b}_1^{*\top}\mathbf{d} = 0$. □

Lemma 5.11.

$$|\Pr[\langle \mathcal{A}, \widehat{\mathrm{H}}_{1,2}^{h_{t-1}}\rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathrm{H}}_{1,3}^{h_{t-1}}\rangle = 1]| \leq \mathrm{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda).$$

The proof of the Lemma 5.11 is almost the same as that of Lemma 5.9.

Lemma 5.12.

$$\Pr[\langle \mathcal{A}, \widehat{\mathrm{H}}_{1,3}^{h_{t-1}}\rangle = 1] = \Pr[\langle \mathcal{A}, \widehat{\mathrm{H}}_{0}^{h_{t}}\rangle = 1].$$

The proof of the Lemma 5.12 is the same as that of Lemma 3.7. From the above observation, Lemma 5.5 holds.

□

# 6 IMPLEMENTATION AND EVALUATION

We implement our KP-ABE and CP-ABE schemes and measure the benchmarks of our schemes on an ordinary personal computer (PC) and two smartphones, iPhone XR and Pixel 3. Their specifications are shown in Table 2. We present theoretical comparisons with previous schemes in Section 7 for reference.

We also implement building blocks of our schemes such as group exponentiation, hashing to $G_1$ and pairing from scratch. For efficiency, our programs are implemented in C and assembly language using major efficient algorithms, e.g., $w$-NAF and GLV/GLS for $G_1$ and $G_2$-exponentiation and the sliding window algorithm for $G_T$-exponentiation. Some functions such as SHA-256 are employed from OpenSSL version 1.1.0j. We also use optimization techniques such as multi-exponentiation and multi-pairing. We use BN curve whose order of groups is a 462-bit prime for pairing groups [5]. This is a new parameter considering the results by Kim et al., who proposed a technique that solves the discrete logarithm problem in a finite field [19, 20]. The running times of time-consuming operations on the PC are listed in Table 3.

As we can see in Fig 5 and 6, the efficiency of KeyGen and Dec in KP-ABE (resp. Enc and Dec in CP-ABE) is affected by formula $f$ used in a secret key (resp. a ciphertext). More concretely, in KeyGen of our KP-ABE and Enc of our CP-ABE, the numbers of exponentiation in $G_1$ and $G_2$ increase proportionally to those of negation and multi-use, respectively. On the other hand, the number of hashing decreases proportionally to that of multi-use. In Dec, the numbers of exponentiation and pairings increase proportionally to the numbers of negation and multi-use, respectively.

To clarify the effects of these factors, we measure benchmarks for the four types of formulae.

(1) no negations and multi-uses (no neg. & no mult.):
    i.e., (Label-1:$v_1$ AND Label-2:$v_2$ AND …),
(2) all negations and no multi-uses (all neg. & no mult.):
    i.e., (Label-1:NOT $v_1$ AND Label-2:NOT $v_2$ AND …),
(3) no negations and all multi-uses (no neg. & all mult.):
    i.e., (Label-1:$v_1$ AND Label-1:$v_1$ AND …),
(4) all negations and multi-uses (all neg. & all mult.):
    i.e., (Label-1:NOT $v_1$ AND Label-1:NOT $v_2$ AND …).

The formula in item 3 is meaningless but just for measuring the effect of multi-use. The reason for not using OR in a formula is to use all elements in a secret key for decryption, which is necessary to evaluate how the number of attributes affects the running time.

We present the benchmarks on the PC in Fig 9 and 10, iPhone XR in Fig 11 and 12, and Pixel 3 in Fig 13 and 14. The figures show the benchmarks with respect to a formula or attribute set with $1, 10, 20, \ldots, 100$ attributes for each case listed above. Enc in KP-ABE and KeyGen in CP-ABE are not affected by the types of formula, and we measure the benchmark for encryption/key generation with attributes Label-1:$v_1, \ldots,$ Label-$n$:$v_n$.

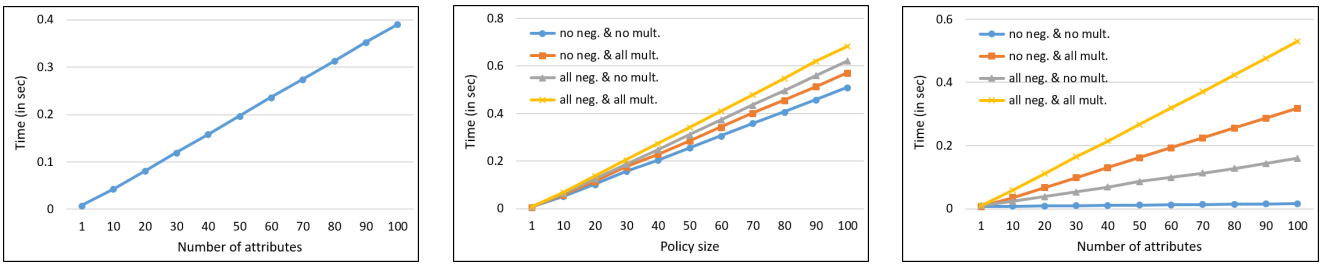
In all cases, our KP-ABE (resp. CP-ABE) scheme takes about 0.4 to 0.7s (resp. 0.4 to 0.9s) for encryption and key generation on the PC to handle 100 attributes. Our schemes allow very fast decryption for a monotone formula without multi-use (item 1), and they take only about 0.02s (KP & CP) for a formula with 100 attributes. We can assume that our schemes allow similarly fast decryption also for a formula in which the ratio of negation and multi-use is small. Even in the slowest case (item 4), it takes about 0.5 (KP) or 0.7s (CP) for decryption.

Because of small computational resource compared with the PC, the smartphones take more time for each algorithm. The benchmarks show that running times on iPhone XR are relatively close to those on the PC, and they are approximately 1.5 times slower. Google Pixel 3 takes further more time and its running times are 3 to 3.5 times as slow as those on the PC.

**Effects of negation and multi-use.** The benchmarks for KeyGen in KP-ABE and Enc in CP-ABE show that both negation and multi-use slow the running time down. It is reasonable that negation slows the running time down because it just increases the number of exponentiation in $G_1$. In contrast, multi-use decreases the number of hashing to $G_1$ whereas it increases that of exponentiation in $G_2$. The benchmarks show that the former effect is smaller than the latter in our implementation. However, multi-use can shorten

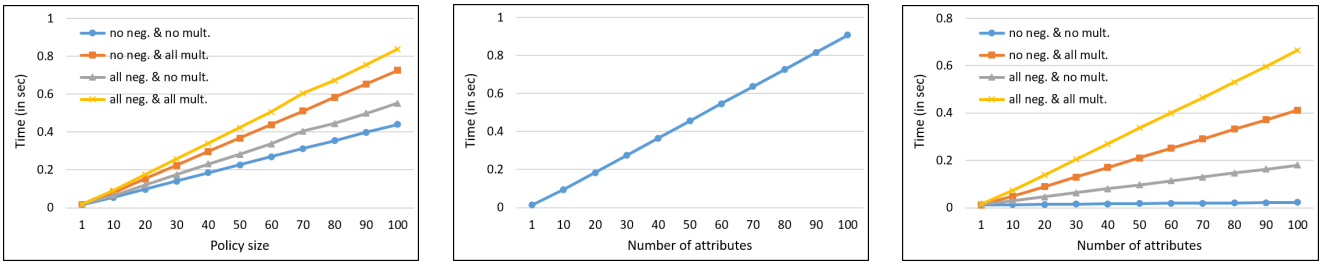
**Table 2: Specifications of devices for our benchmarks.**

| Device | OS | CPU / SoC | Compiler |
|---|---|---|---|
| PC | Ubuntu 18.04.2 LTS | Intel Core i7-8700 @ 3.2 GHz (up to 4.6 GHz by TurboBoost) | gcc 7.4 |
| Apple iPhone XR | iOS 12.2 | Apple A12 Bionic | Apple LLVM 10.0.1 |
| Google Pixel 3 | Android 9 | Qualcomm Snapdragon 845 | Android clang 8.0.2 |



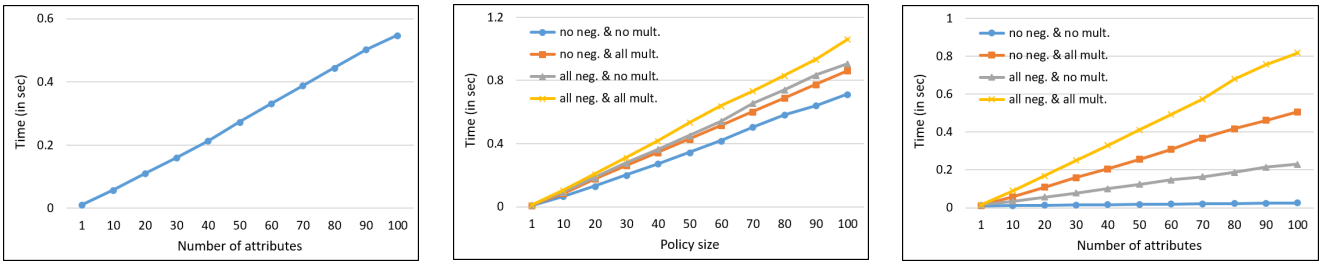

(a) Enc (b) KeyGen (c) Dec

**Figure 9: Benchmarks for KP-ABE on the personal computer.**




(a) Enc (b) KeyGen (c) Dec

**Figure 10: Benchmarks for CP-ABE on the personal computer.**



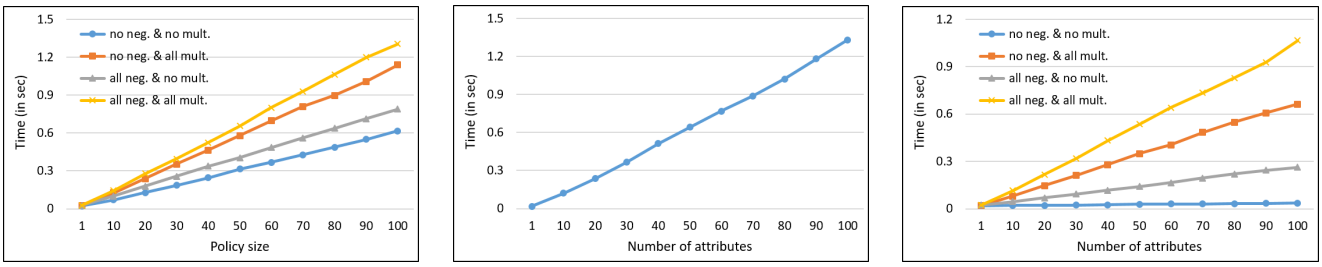

(a) Enc (b) KeyGen (c) Dec

**Figure 11: Benchmarks for KP-ABE on iPhone XR.**

the running time in a platform where exponentiation in $G_2$ is more efficient or hashing to $G_1$ is less efficient.

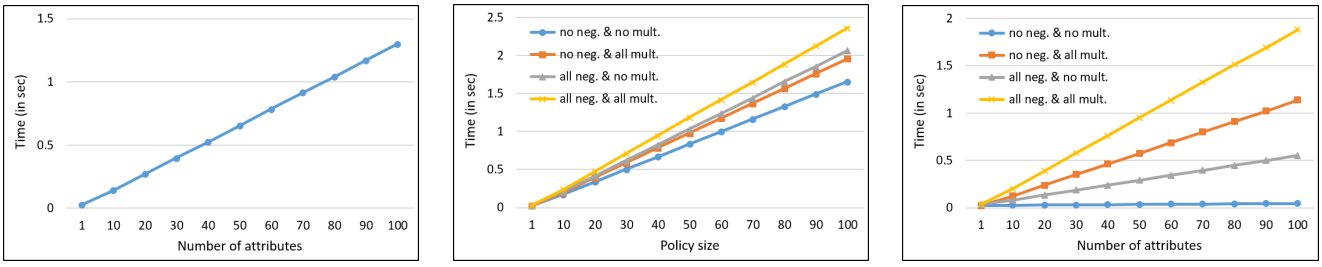
In Dec, both negation and multi-use extend the running time, and the effect of multi-use is larger. This is natural because the number of negation affects that of exponentiation in $G_1$ whereas the number of multi-use affects that of heavier pairings.

**Further optimization.** Our schemes utilize the hash function to $G_1$ in Enc and KeyGen to obtain unboundedness. The benchmarks show the running times with hashing to $G_1$. However, we consider that the evaluation of the hash function is needed only when we add new labels to the system. This is because labels that are used in regular operation are typically fixed. Thus, one can precompute hash values and matrix multiplications and store them in a memory. This optimization will allow faster encryption and key generation.



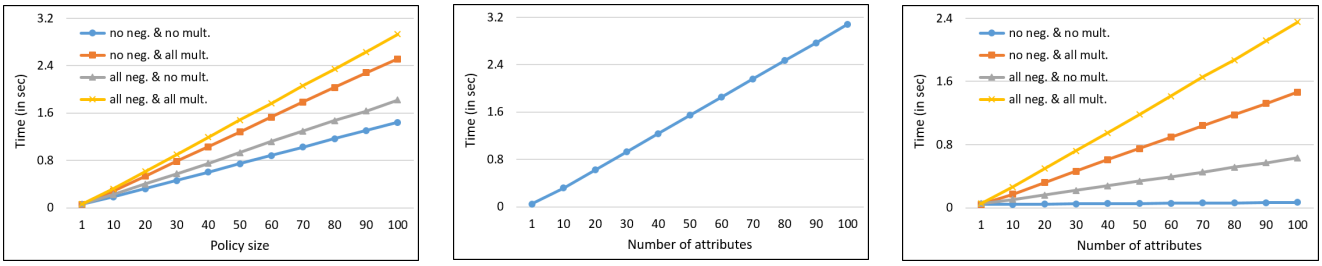

(a) Enc (b) KeyGen (c) Dec

**Figure 12: Benchmarks for CP-ABE on iPhone XR.**




(a) Enc (b) KeyGen (c) Dec

**Figure 13: Benchmarks for KP-ABE on Pixel 3.**




(a) Enc (b) KeyGen (c) Dec

**Figure 14: Benchmarks for CP-ABE on Pixel 3.**

**Table 3: Running times of time-consuming operations on the PC.**

| Operation | Timing (in milli-sec) |
|---|---|
| $G_1$-exponentiation | 0.368 |
| $G_2$-exponentiation | 0.641 |
| $G_T$-exponentiation | 1.950 |
| Hashing to $G_1$ | 0.096 |
| Pairing | 2.080 |

## 7 THEORETICAL COMPARISON

We give theoretical comparisons with some KP-ABE schemes in Tables 4 to 7. That is, we compare them by the number of operations and group elements. For the comparison, we select AC17 by Agrawal and Chase [1], the basic scheme of OT12 by Okamoto and Takashima [27], and the asymmetric variant of GPSW06 by Goyal et al. [16] (written in the FAME paper [1]). The selection criteria are as follows:

- FAME is the most efficient KP-ABE scheme that satisfies properties from (1) to (5) written in Section 1.
- OT12 satisfies unboundedness and can treat the natural negation form (denoted by OT-negation in Section 1.1).
- GPSW06 is the most efficient KP-ABE scheme though it satisfies none of the adaptive security, unboundedness, large universe, fast decryption, and non-monotonicity.

Note that AC17 and OT12 do not satisfy the multi-use property. We consider 2-Lin family $\mathcal{L}_2$ described in Section 4.1 for ours. In Tables 5 and 7, we omit target group elements that hide messages in ciphertexts in these tables since they are not dominant factors (as Agrawal and Chase did [1]). We also omit the number of the multiplication operation in Tables 4 to 6 since it is not a dominant factor compared with exponentiation and pairing operations. The parameters are as follows:

**Table 4: Comparison of key generation algorithms in KP-ABE schemes.**

| schemes | Key generation | | | |
|---|---|---|---|---|
| | $G_1$ | | $G_2$ | |
| | Exp | Hash | Exp | Hash |
| Ours | $15n_t + 18n_f$ | $12n'$ | $3d$ | - |
| AC17 | $9n_1 + 3n_2 + 3$ | $6(n_1 + n_2)$ | 3 | - |
| OT12 | - | - | $84n_1 + 15$ | - |
| GPSW06 | - | - | $n_1$ | - |

**Table 5: Comparison of encryption algorithms in KP-ABE schemes.**

| schemes | Encryption | | | |
|---|---|---|---|---|
| | $G_1$ | | $G_2$ | |
| | Exp | Hash | Exp | Hash |
| Ours | $12m$ | $12m$ | 3 | - |
| AC17 | $6m$ | $6m$ | 3 | - |
| OT12 | $84m + 15$ | - | - | - |
| GPSW06 | $m$ | - | - | - |

**Table 6: Comparison of decryption alogrithms in KP-ABE schemes. We omit multiplication costs in $G_1, G_2, G_T$ since they are tiny comparing with exponentiation and pairing.**

| Schemes | Decryption | | | |
|---|---|---|---|---|
| | Exponentiation | | | Pairing |
| | $G_1$ | $G_2$ | $G_T$ | |
| Ours | $9I_f d$ | - | - | $6d$ |
| AC17 | - | - | - | 6 |
| OT12 | $I_f$ | - | - | $14I + 5$ |
| GPSW06 | - | - | - | $I$ |

**Table 7: Size comparison of KP-ABE schemes.**

| Schemes | Key size | | Ciphertext size | |
|---|---|---|---|---|
| | $G_1$ | $G_2$ | $G_1$ | $G_2$ |
| Ours | $3(n_t + 2n_f)$ | $3d$ | $3m$ | 3 |
| AC17 | $3n_1$ | 3 | $3m$ | 3 |
| OT12 | - | $14n_1 + 5$ | $14m + 5$ | - |
| GPSW06 | - | $n_1$ | $m$ | - |

- $d$: the maximum number of multi-use.
- $n, n_t, n_f$: the number of inputs, non-negated and negated inputs to a policy, respectively ($n = n_t + n_f$).
- $n'$: the number of distinct labels ($n' \leq n$).
- $n_1, n_2$: the number of rows and columns of a matrix for span programs.
- $m$: the number of attributes.
- $I, I_t, I_f$: the number of attributes, non-negated and negated attributes in decryption, respectively ($I = I_t + I_f$).

GPSW06 is the most efficient (note that this is obvious since the functionality of GPSW06 is limited). Ours is much more efficient than OT12. Note that hashing to $G_1$ is not an expensive operation as we saw in Section 6. Thus, we focus on a comparison with AC17 below.

We show the number of operations in algorithm KeyGen in Table 4. If we consider $d = 1$ (no multi-use), then the efficiency in $G_2$ of ours is the same as that of AC17. Regarding $G_1$, ours is about 2 times slower than AC17 (note that $15n_t + 18n_f = 15n + 3n_f$).

We show the number of operations in algorithm Enc in Table 5. Ours is just 2 times slower than AC17 in $G_1$.

We show the number of operations in algorithm Dec in Table 6. It is easy to see that if we use neither negation nor duplicate attributes, then the performance of ours the same as that of AC17. As we saw in Section 6, if we use many negations and duplicate attributes, then our decryption algorithm gets slower.

We show the number of group elements in each secret key and ciphertext in Table 7. It is easy to see that if we use neither negation nor duplicate attributes, then the performance of ours is the same as that of AC17. Even if we use negation, the number of group elements increases only $3n_f$ elements in $G_1$ compared with AC17 since $3(n_t + 2n_f) = 3(n + n_f)$ (due to $n = n_t + n_f$). Again, we stress that AC17 cannot treat negation and multi-use of attributes.

Overall, ours is a little bit less efficient than AC17 in the theoretical sense. However, ours is more expressive than AC17 since ours can treat natural negation and multi-use. Moreover, our implementation is efficient enough for practical use as we saw in Section 6.

## REFERENCES


[1] Shashank Agrawal and Melissa Chase. 2017. FAME: Fast Attribute-based Message Encryption. In *ACM CCS 17*, Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM Press, 665–682. https://doi.org/10.1145/3133956.3134014

[2] Shashank Agrawal and Melissa Chase. 2017. Simplifying Design and Analysis of Complex Predicate Encryption Schemes. In *EUROCRYPT 2017, Part I (LNCS)*, Jean-Sébastien Coron and Jesper Buus Nielsen (Eds.), Vol. 10210. Springer, Heidelberg, 627–656. https://doi.org/10.1007/978-3-319-56620-7_22

[3] Nuttapong Attrapadung. 2019. Unbounded Dynamic Predicate Compositions in Attribute-Based Encryption. In *EUROCRYPT 2019, Part I (LNCS)*, Vincent Rijmen and Yuval Ishai (Eds.). Springer, Heidelberg, 34–67. https://doi.org/10.1007/978-3-030-17653-2_2

[4] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. 2011. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In *PKC 2011 (LNCS)*, Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi (Eds.), Vol. 6571. Springer, Heidelberg, 90–108. https://doi.org/10.1007/978-3-642-19379-8_6

[5] Razvan Barbulescu and Sylvain Duquesne. 2017. Updating key size estimations for pairings. Cryptology ePrint Archive, Report 2017/334. (2017). http://eprint.iacr.org/2017/334.

[6] Mihir Bellare and Phillip Rogaway. 1995. Optimal Asymmetric Encryption. In *EUROCRYPT'94 (LNCS)*, Alfredo De Santis (Ed.), Vol. 950. Springer, Heidelberg, 92–111. https://doi.org/10.1007/BFb0053428

[7] Mihir Bellare and Phillip Rogaway. 1996. The Exact Security of Digital Signatures: How to Sign with RSA and Rabin. In *EUROCRYPT'96 (LNCS)*, Ueli M. Maurer (Ed.), Vol. 1070. Springer, Heidelberg, 399–416. https://doi.org/10.1007/3-540-68339-9_34

[8] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-Policy Attribute-Based Encryption. In *2007 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 321–334. https://doi.org/10.1109/SP.2007.11

[9] Dan Boneh and Matthew K. Franklin. 2001. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001 (LNCS)*, Joe Kilian (Ed.), Vol. 2139. Springer, Heidelberg, 213–229. https://doi.org/10.1007/3-540-44647-8_13

[10] Ran Canetti, Oded Goldreich, and Shai Halevi. 1998. The Random Oracle Methodology, Revisited (Preliminary Version). In *30th ACM STOC*. ACM Press, 209–218. https://doi.org/10.1145/276698.276741

[11] Jie Chen, Romain Gay, and Hoeteck Wee. 2015. Improved Dual System ABE in Prime-Order Groups via Predicate Encodings. In *EUROCRYPT 2015, Part II (LNCS)*, Elisabeth Oswald and Marc Fischlin (Eds.), Vol. 9057. Springer, Heidelberg, 595–624. https://doi.org/10.1007/978-3-662-46803-6_20

[12] Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. 2018. Unbounded ABE via Bilinear Entropy Expansion, Revisited. In *EUROCRYPT 2018, Part I (LNCS)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.), Vol. 10820. Springer, Heidelberg, 503–534. https://doi.org/10.1007/978-3-319-78381-9_19

[13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. 2017. An Algebraic Framework for Diffie-Hellman Assumptions. *Journal of Cryptology* 30, 1 (Jan. 2017), 242–288. https://doi.org/10.1007/s00145-015-9220-6

[14] Eiichiro Fujisaki and Tatsuaki Okamoto. 2013. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *Journal of Cryptology* 26, 1 (Jan. 2013), 80–101. https://doi.org/10.1007/s00145-011-9114-1

[15] Junqing Gong, Xiaolei Dong, Jie Chen, and Zhenfu Cao. 2016. Efficient IBE with Tight Reduction to Standard Assumption in the Multi-challenge Setting. In *ASIACRYPT 2016, Part II (LNCS)*, Jung Hee Cheon and Tsuyoshi Takagi (Eds.), Vol. 10032. Springer, Heidelberg, 624–654. https://doi.org/10.1007/978-3-662-53890-6_21

[16] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM CCS 06*, Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati (Eds.). ACM Press, 89–98. https://doi.org/10.1145/1180405.1180418 Available as Cryptology ePrint Archive Report 2006/309.

[17] Zahra Jafargholi, Chethan Kamath, Karen Klein, Ilan Komargodski, Krzysztof Pietrzak, and Daniel Wichs. 2017. Be Adaptive, Avoid Overcommitting. In *CRYPTO 2017, Part I (LNCS)*, Jonathan Katz and Hovav Shacham (Eds.), Vol. 10401. Springer, Heidelberg, 133–163. https://doi.org/10.1007/978-3-319-63688-7_5

[18] Jonathan Katz, Amit Sahai, and Brent Waters. 2008. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *EUROCRYPT 2008 (LNCS)*, Nigel P. Smart (Ed.), Vol. 4965. Springer, Heidelberg, 146–162. https://doi.org/10.1007/978-3-540-78967-3_9

[19] Taechan Kim and Razvan Barbulescu. 2016. Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case. In *CRYPTO 2016, Part I (LNCS)*, Matthew Robshaw and Jonathan Katz (Eds.), Vol. 9814. Springer, Heidelberg, 543–571. https://doi.org/10.1007/978-3-662-53018-4_20

[20] Taechan Kim and Jinhyuck Jeong. 2017. Extended Tower Number Field Sieve with Application to Finite Fields of Arbitrary Composite Extension Degree. In *PKC 2017, Part I (LNCS)*, Serge Fehr (Ed.), Vol. 10174. Springer, Heidelberg, 388–408. https://doi.org/10.1007/978-3-662-54365-8_16

[21] Lucas Kowalczyk and Hoeteck Wee. 2019. Compact Adaptively Secure ABE for $\mathsf{NC}s^1$ from $k$-Lin. In *EUROCRYPT 2019, Part I (LNCS)*, Vincent Rijmen and Yuval Ishai (Eds.). Springer, Heidelberg, 3–33. https://doi.org/10.1007/978-3-030-17653-2_1

[22] Allison Lewko and Brent Waters. 2010. Decentralizing Attribute-Based Encryption. Cryptology ePrint Archive, Report 2010/351. (2010). http://eprint.iacr.org/2010/351.

[23] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. 2010. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *EUROCRYPT 2010 (LNCS)*, Henri Gilbert (Ed.), Vol. 6110. Springer, Heidelberg, 62–91. https://doi.org/10.1007/978-3-642-13190-5_4

[24] Allison B. Lewko, Amit Sahai, and Brent Waters. 2010. Revocation Systems with Very Small Private Keys. In *2010 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 273–285. https://doi.org/10.1109/SP.2010.23

[25] Allison B. Lewko and Brent Waters. 2010. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010 (LNCS)*, Daniele Micciancio (Ed.), Vol. 5978. Springer, Heidelberg, 455–479. https://doi.org/10.1007/978-3-642-11799-2_27

[26] Tatsuaki Okamoto and Katsuyuki Takashima. 2010. Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In *CRYPTO 2010 (LNCS)*, Tal Rabin (Ed.), Vol. 6223. Springer, Heidelberg, 191–208. https://doi.org/10.1007/978-3-642-14623-7_11

[27] Tatsuaki Okamoto and Katsuyuki Takashima. 2012. Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In *ASIACRYPT 2012 (LNCS)*, Xiaoyun Wang and Kazue Sako (Eds.), Vol. 7658. Springer, Heidelberg, 349–366. https://doi.org/10.1007/978-3-642-34961-4_22

[28] Rafail Ostrovsky, Amit Sahai, and Brent Waters. 2007. Attribute-based encryption with non-monotonic access structures. In *ACM CCS 07*, Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson (Eds.). ACM Press, 195–203. https://doi.org/10.1145/1315245.1315270

[29] Amit Sahai and Brent R. Waters. 2005. Fuzzy Identity-Based Encryption. In *EUROCRYPT 2005 (LNCS)*, Ronald Cramer (Ed.), Vol. 3494. Springer, Heidelberg, 457–473. https://doi.org/10.1007/11426639_27

[30] Brent Waters. 2009. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *CRYPTO 2009 (LNCS)*, Shai Halevi (Ed.), Vol. 5677. Springer, Heidelberg, 619–636. https://doi.org/10.1007/978-3-642-03356-8_36

[31] Hoeteck Wee. 2014. Dual System Encryption via Predicate Encodings. In *TCC 2014 (LNCS)*, Yehuda Lindell (Ed.), Vol. 8349. Springer, Heidelberg, 616–637. https://doi.org/10.1007/978-3-642-54242-8_26

[32] Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. 2014. A Framework and Compact Constructions for Non-monotonic Attribute-Based Encryption. In *PKC 2014 (LNCS)*, Hugo Krawczyk (Ed.), Vol. 8383. Springer, Heidelberg, 275–292. https://doi.org/10.1007/978-3-642-54631-0_16