Fast, Compact, and Expressive Attribute-Based Encryption

Junichi Tomida*
NTT Corporation, Japan

Yuto Kawahara†
NTT Corporation, Japan

Ryo Nishimaki‡
NTT Corporation, Japan

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 adaptively secure under a standard assumption, simultaneously. Our schemes are more expressive than previous schemes and efficient enough. To achieve the adaptive security along with the other properties, we refine the technique introduced by Kowalczyk and Wee (Eurocrypt'19) so that we can apply the technique more expressive ABE schemes. Furthermore, we also present a new proof technique that allows us to remove redundant elements used in their ABE schemes. We 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) [18] 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 [10], threshold [31], orthogonality of vectors [20], and so on. One of the most notable relations among them is that expressed by an access structure [8,18]. 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.

Recently, Agrawal and Chase proposed practical 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);

*junichi.tomida.vw@hco.ntt.co.jp
†yuto.kawahara.yk@hco.ntt.co.jp
‡ryo.nishimaki.zk@hco.ntt.co.jp

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 allow us to use

6. negation *in a natural form* (non-monotonicity);

7. the same attribute more than once (multi-use of attributes or compactness);

in a policy. These properties allow us to use more fine-grained policies that are commonly used in practice. Negation is essential for access control by blacklisting. Multi-use of attributes in policies is indispensable to express certain types of policies such as (A AND B) OR (A AND C) OR (B AND D), where A, B, C, D are Boolean variables.

Thanks to great works on ABE [3,23,29], 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 [12], 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, 16].

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, 26, 28–30, 34]. Among them, only the negation form defined by Okamoto and Takashima (OT negation) [28, 29] is different from that by the others (non-OT negation). Considering an example is the best way to describe the difference. Let attributes consist of a pair of a label and value, e.g., Year:1991-2000, where Year is a label and 1991-2000 is a value. Suppose there are two labels Year and Category in an access control system supported by KP-ABE. 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 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, 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

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

Scheme	Unbounded-ness	Large universe	Type-III	Fast Dec	Standard assump.	Non-monotonicity	Multi-use	w/o RO
OT12 [29]	✓	✓	✓	×	✓	✓	×	✓
AC17 [1]	✓	✓	✓	✓	✓	×	×	×
CGKW18 [14]	✓	✓	✓	×	✓	×	×	✓
KW19 [23]	✓	✓	✓	×	✓	×	✓	✓
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.

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.

Note that we can use monotone ABE as non-monotone ABE by preparing attributes for both positive and negative if they are small-universe constructions, in which the number of attributes are polynomially bounded. That is, non-possession of attributes can be expressed by possession of negative attributes. However, this is not the case in large-universe constructions because we cannot attach an exponentially large number of negative attributes to a ciphertext or secret key. Hence, monotone ABE and non-monotone ABE are completely different things in the context of large-universe constructions.

Multi-use of attributes (Compactness). Many ABE schemes whose security relies on the dual system methodology [32] have a one-use restriction on access structures [13, 14, 25, 28, 29]. 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 nominal attributes for each single attribute in advance like CATEGORY:jazz-1, ..., CATEGORY:jazz-d for CATEGORY:jazz. However, this solution has two problems. The first is that the maximum number d of multi-use is fixed 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 of multi-use, and it leads to efficiency loss. This 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 satisfies "compactness" [23].

1.2 Design of Our ABE Schemes

In the following, we focus on the design our KP-ABE scheme, and the CP-ABE scheme is similarly constructed. The relation R of our ABE is close to that by Okamoto and Takashima in [29]. As we mentioned, an attribute consists of a label and value. Our schemes are unbounded in terms of this tuple, that is, a user can attach an arbitrary number of tuples of a label and value to a ciphertext or secret key. 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)). That is, our scheme A formal definition of R is described in Definition 2.5.

Our scheme is based on the dual system encryption, which we can instantiate from either composite-order or prime-order bilinear groups [13,27,32,33]. Our actual scheme is based on prime-order bilinear groups following the framework by Chen et al. [13] 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 here. 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 $R : \{0,1\}^* \to G_1 \times G_1$ be a hash function modeled as a random oracle, and its input is a label. We denote the output of $R(i)$ by $(g_1^{u_i}, g_1^{h_i})$. Then, our scheme can be written as

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

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, α_i is a share of the secret α, 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 the construction is a combination of secret sharing (SS) and two-mode identity-based encryption (TIBE) [34]. TIBE is obtained by just combining identity-based encryption (IBE) and negation of IBE (NIBE). Our scheme can instantiate an arbitrary number of TIBE on the fly by leveraging hash function R, and each instance corresponds to each label. A secret key of our scheme consists of secret keys of IBE and NIBE, and each secret key hides a share α_i of a master secret α generated by SS according to the formula. A ciphertext of ABE consists of ciphertexts of IBE, which have the same form as those in Boneh-Boyen IBE [9]. Note that ciphertexts of IBE and NIBE are identical, and thus we do not need to include both ciphertexts of IBE and NIBE in a ciphertext of our scheme. In decryption, one computes $\{e(g_1,h_1)^{s\alpha_i}\}_i$ for labels in which the relation of (in)equality between the ciphertext and secret keys is satisfied. Note that one cannot compute $e(g_1,h_1)^{s\alpha_i}$ if the relation of (in)equality does not hold in label i, thanks to the security of underlying TIBE. If $e(g_1,h_1)^{s\alpha}$ 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, $e(g_1,h_1)^{s\alpha_i}$ cannot be computed if a ciphertext of ABE does not contain a ciphertext of TIBE for label i, and this property yields OT negation.

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 [33] if ψ is injective, or the scheme has the one-use restriction of labels in policies. 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) [23]. More precisely, they proposed a methodology to prove the adaptive security of the most simple ABE scheme, which supports monotone NC_1 circuits (or equivalently Boolean formulae) for a small attribute universe. The scheme can be written in composite-order groups as

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

Roughly speaking, this scheme can be seen as KP-ABE whose ingredients are ElGamal-like encryption whereas the counterpart of our scheme corresponds to TIBE.

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:

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

where γ_i is a share of a random secret γ.

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:

$$\left\{\begin{array}{l} \mathsf{ct} : (g_2^{s}, \{g_2^{sw_i}\}_{i\in S}), \\ \\ \mathsf{sk} : (\{h_2^{r_i}\}_{i\in[n]}, \{h_2^{r_i w_{\psi(i)} + \boxed{\gamma_{0,i}}}\}_{i\in[n]}) \end{array}\right\} \approx_c \left\{\begin{array}{l} (g_2^{s}, \{g_2^{sw_i}\}_{i\in S}), \\ \\ (\{h_2^{r_i}\}_{i\in[n]}, \{h_2^{r_i w_{\psi(i)} + \boxed{\gamma_{1,i}}}\}_{i\in[n]}) \end{array}\right\}$$

where $\gamma_{0,i}$ is a share of secret 0 and $\gamma_{1,i}$ is a share of secret γ. 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 output that has sufficient information to embed the problem into sk. Simultaneously, the part must be so small that the simulator can guess it with non-negligible probability. In our case, the part tells the correct element in sk where the simulator embeds the problem. Observe that each γ_i is masked by ElGamal-like 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 the ElGamal-like encryption, that is, public-key encryption.

To overcome the problem, we introduce new usage of KW19 framework 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 TIBE 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:

$$\left\{\begin{array}{l}(g^s, \{g^{sw_i}\}_{i\in S}),\\ \\ (\{h_1^{r_i}\}_{i\in[n]}, \{h_1^{r_i w_{\psi(i)}} \cdot h_2^{\boxed{\gamma_{0,i}}}\}_{i\in[n]})\end{array}\right\} \approx_c \left\{\begin{array}{l}(g^s, \{g^{sw_i}\}_{i\in S}),\\ \\ (\{h_1^{r_i}\}_{i\in[n]}, \{h_1^{r_i w_{\psi(i)}} \cdot h_2^{\boxed{\gamma_{1,i}}}\}_{i\in[n]})\end{array}\right\}.$$

The difference from the original core 1-ABE is that our core 1-ABE considers both normal space (G_1 and H_1) and semi-functional space (G_2 and H_2), whereas the original one considers only semi-functional space. We use the dual system methodology to randomize the component in H_2. Let i^* be the location where γ_{i^*} is supposed to be changed in some two hybrids, which means that $i^* \notin S$. Then, from the subgroup 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 γ_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.

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 $d = \max \pi(i)$ is the maximum number of multi-use of labels in the policy. 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. We show that we can prove the security of our schemes under standard assumptions even if we use this construction. Note that the construction by Agrawal and Chase relies on a q-type assumption.

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. [19], 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]. We show that we can utilize a pseudorandom function (PRF) to significantly ease the security proof. Concretely, we can skip the part that corresponds to 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 [14, 17] due to a technical reason.

1.5 Implementation and Evaluation

We implement our KP and CP-ABE schemes in 128-bit security level and measure benchmarks for an ordinary personal computer and two smartphones: iPhone XR and Pixel 3. 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.

We also implement KP and CP-ABE schemes by Okamoto and Takashima (OT12), which are the only known ABE schemes that support OT negation and the unboundedness [29]. There are no known schemes that are as expressive as ours (see Table 1), and OT12 seems to have a closet functionality. This is why we choose OT12 to compare. The comparison between our schemes and OT12 shows that our schemes achieve significant speedups for each algorithm.

2 Preliminaries

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, ℓ) 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

Boolean Formula and NC^1. A monotone Boolean formula can be represented by a Boolean circuit whose all gates have fan-in 2 and fan-out 1. We can specify a monotone Boolean formula $f : \{0, 1\}^n \to \{0, 1\}$ as $f = (n, w, v, G)$, where $n, m, v \in \mathbb{N}$ and $G : [v] \to \{\text{AND, OR}\} \times [w]^3$. This means the Boolean formula f has n input wires, w wires including the input wires, and v gates. We number the wires $1, \ldots, w$ and the gates $1, \ldots, v$. The function G specifies a type, incoming wires, and an outgoing wire

of each gate. That is, for $G(i) = (T, a, b, c)$ such that $a < b < c$, T specifies a type of gate i, a and b specify the incoming wires, and c specifies the outgoing wire. A non-monotone Boolean formula additionally contains NOT gates, which have fan-in 1 and fan-out 1. It is well-known that we can express all non-monotone Boolean formulae by one in which all NOT gates are put on the input wires, and we only consider such formulae in this paper. Thus, we can specify a non-monotone Boolean formula $f' : \{0,1\}^n \to \{0,1\}$ as $f' = (f, t)$, where $f = (n, w, v, G)$ is a monotone Boolean formula and $t : [n] \to \{0,1\}$ specifies input gates that connect to a NOT gate. That is, input wire i connects to a NOT gate if $t(i) = 0$ and does not if $t(i) = 1$.

Standard complexity theory tells us that circuit complexity class NC^1 and Boolean formulae are equivalent. It is known also that NC^1 is equivalent to the class captured by log-depth Boolean formulae (see e.g., [23]). Thus, the circuit complexity class captured by Boolean formulae is equivalent to the class captured by log-depth Boolean formulae.

Definition 2.1 (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 2.2 (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 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^λ and outputs a description of bilinear groups $\mathbb{G}$ with $\Omega(\lambda)$ 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 [15], 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 2.3 ($\mathcal{D}_{j,k}$-MDDH Assumption)**.** *For $j > k$, let $\mathcal{D}_{j,k}$ be a matrix distribution over $\mathbb{Z}_p^{j \times k}$ that outputs full rank matrix with overwhelming probability. 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 $\mathcal{A}$,

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

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 [15]. Thus, in asymmetric bilinear groups, we can utilize the bilateral $\mathcal{D}_k$-MDDH assumption for $k \geq 2$.

Matrix Notation. For a matrix $\mathbf{A} \in \mathcal{D}_k$, we define a matrix $\mathbf{A}^*$ and vectors $\mathbf{a}_1$ and $\mathbf{a}_1^*$ as follows. Vector $\mathbf{a}_1$ is a $k+1$ dimensional vector whose last entry is 1 and the others are 0. Then, it is not hard to see that $\overline{\mathbf{A}} := (\mathbf{A}||\mathbf{a}_1)$ forms a basis of $\mathbb{Z}_p^{k+1}$ because the first k rows of a matrix $\mathbf{A}$ chosen from $\mathcal{D}_k$ form an invertible matrix. $\mathbf{A}^*$ and $\mathbf{a}_1^*$ 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 we have $\mathbf{A}^\top\mathbf{A}^* = \mathbf{I}_k$, $\mathbf{A}^\top\mathbf{a}_1^* = \mathbf{0}$, and $\mathbf{A}^*\mathbf{A}^\top + \mathbf{a}_1^*\mathbf{a}_1^\top = \mathbf{I}_{k+1}$. We use a similar notation for a matrix $\overline{\mathbf{B}} \in \mathsf{GL}_{k+\eta}(\mathbb{Z}_p)$ where $\eta \in \mathbb{N}$. $\mathbf{B}$ and $\mathbf{b}_i$ denote a matrix consists of the first k columns of $\overline{\mathbf{B}}$ and a vector consists of the $k+i$-th column of $\overline{\mathbf{B}}$, respectively. Similarly, $\mathbf{B}^*$, $\mathbf{b}_i^*$ denote a matrix consists of the first k columns of $(\overline{\mathbf{B}}^\top)^{-1}$ and a vector consists of the $k+i$-th 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.

2.3 Attribute-Based Encryption

Definition 2.4 (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^λ 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 ct_x.*

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

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

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*

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

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 λ for all stateful PPT adversary $\mathcal{A}$:*

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

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

A relation 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 certain components in an 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 [29], 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.

Definition 2.5 (Relation R)**.** *Relations R_{KP} and R_{CP} for our KP and CP-ABE schemes, respectively, are defined as follows. Let $R : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a relation defined as follows:*

- $\mathcal{X} = \bigcup_{i\in\mathbb{N}} \mathbb{Z}_p^i \times \Phi_i$*, where Φ_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 Ψ_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 $b_i := \begin{cases} t(i) \odot \mathrm{true}(x_{\phi^{-1}(\psi(i))} = y_i) & \psi(i) \subseteq \mathrm{Im}(\phi) \\ 0 & \psi(i) \not\subseteq \mathrm{Im}(\phi) \end{cases}$, where $\odot$ denotes xnor. Then, $R(x,y) = 1 \Leftrightarrow f(b) = 1$.*

Then, $R_{\mathsf{KP}} : \mathcal{X}_{\mathsf{KP}} \times \mathcal{Y}_{\mathsf{KP}} \to \{0,1\}$ is defined as $\mathcal{X}_{\mathsf{KP}} := \mathcal{X}$, $\mathcal{Y}_{\mathsf{KP}} := \mathcal{Y}$, and $R_{\mathsf{KP}}(x,y) = R(x,y)$, whereas $R_{\mathsf{CP}} : \mathcal{X}_{\mathsf{CP}} \times \mathcal{Y}_{\mathsf{CP}} \to \{0,1\}$ is defined as $\mathcal{X}_{\mathsf{CP}} := \mathcal{Y}$, $\mathcal{Y}_{\mathsf{CP}} := \mathcal{X}$, and $R_{\mathsf{CP}}(x,y) = R(y,x)$

For $\mathcal{X}$, each element of $\mathbf{x} \in \mathbb{Z}_p^m$ corresponds to a value for some label, and ϕ 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 where $H_1 : \{0,1\}^* \to \mathbb{Z}_p$ is a collision resistant hash function.

For $\mathcal{Y}$, each element of $\mathbf{y} \in \mathbb{Z}_p^n$ corresponds to the value for each input wire of f, and ψ specifies which label each input wire of f is associated with. Additionally, t specifies whether each input wire connects to a NOT gate. For instance, let us consider a predicate (Age:25 AND Hobby:NOT baseball). Then, $y = (\mathbf{y}, f, \psi, t)$ can be set as $\mathbf{y} := (25, H_1(\text{baseball}))$, f is a formula with a single AND gate, $\psi(1) :=$ Age and $\psi(2) :=$ Hobby, and $t(1) = 1$ and $t(2) = 0$.

Definition 2.6 (Linear Secret Sharing Scheme)**.** *A linear secret sharing scheme (LSSS) for a function class $\mathcal{F}$ consists of two algorithms* Share *and* Rec*.*

$\mathsf{Share}(f, \mathbf{k})$**:** *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 $\mathbf{k}_1, \ldots, \mathbf{k}_n \in \mathbb{Z}_p^\ell$.*

$\mathsf{Rec}(f, x, \{\mathbf{k}_i\}_{x_i=1})$**:** *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 [23], which is based on the framework by Jafargholi et al. [19]. The framework helps us to prove adaptive security of cryptographic schemes that are selectively secure.

Definition 2.7 (Interactive Game)**.** *An interactive game G is a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In the game, $\mathcal{A}$ and $\mathcal{C}$ send messages interactively, and the messages sent by $\mathcal{C}$ 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 $\langle \mathcal{A}, \mathsf{G} \rangle$.*

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 $\widehat{\mathsf{G}}$, 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 G_0 and G_1 are computationally indistinguishable, i.e.,

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

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

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

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

Lemma 2.1 (Adaptive Security Lemma [23])**.** *Let G_0 and G_1 be adaptive interactive games and $\{\widehat{\mathsf{H}}^{h_i}\}_{0 \leq i \leq L}$ be selective hybrids defined above. Suppose they satisfy the two properties:*

- *$\mathsf{G}_0 = \mathsf{H}^{h_0}$ and $\mathsf{G}_1 = \mathsf{H}^{h_L}$, where H^{h_0} and H^{h_L} are the same as $\widehat{\mathsf{H}}^{h_0}$ and $\widehat{\mathsf{H}}^{h_L}$, respectively, except that $\mathcal{A}$ does not declare z at the beginning. Note that $\mathcal{C}$'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*

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

Then, we have

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

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 2.8 (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 (the gate with the output wire), 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 input wire i whose input corresponds to 0, i.e., $b_i = 0$.*
2. *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.*
3. *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.*

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 2.9 (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 pebbling game.* R'*-bit string* r_ι *specifies the configuration at the* ι*-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, and 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 2.2 (Pebbling Lemma [23])**.** *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 Our KP-ABE Scheme

First, we describe a linear secret sharing scheme that we use in our schemes as a building block.

3.1 Linear Secret Sharing for Boolean Formulae

Our secret sharing scheme for monotone Boolean formulae is described in Fig 1, which is essentially the same as the scheme in [24, Appendix G]. Note 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 their framework even if we replace the secret sharing scheme to ours.

$\underline{\mathsf{Share}(f, \mathbf{k})}$
Input: A monotone Boolean formula $f = (n, w, v, G)$ 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 use the following lemma on the secret sharing scheme in the security proof of our scheme.

Lemma 3.1. *Let* Share *be the algorithm defined in Fig 1. For all* $\ell, n \in \mathbb{N}$*, monotone Boolean formulae* $f = (n, w, v, G)$*,* $\mathbf{k}, \mathbf{a} \in \mathbb{Z}_p^\ell$*, 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 $\mathbf{z}_i$ for $i \in [w]$ be values set on a wire i in the execution of $\mathsf{Share}(f, \mathbf{z})$. From the procedure of the scheme, we have $\mathbf{z}_i = b_{\mathsf{out}}\mathbf{z} + \sum_{g \in S} b_g \mathbf{u}_g$ for some subset S of all gates in f, $b_{\mathsf{out}} \in \{0, 1\}$, and $b_g \in \{-1, 1\}$. Note that S, b_{out}, b_g are determined by f and i.

Let $\mathbf{k}_i$, $\mathbf{k}'_i$, and σ_i for $i \in [w]$ 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. Defining $\mathbf{u}_g := \mathbf{u}'_g + u_g\mathbf{a}$ does not change the joint distribution of all $\mathbf{u}_g$. In this case, we have $\mathbf{k}_i = \mathbf{k}'_i + \mu_i\mathbf{a}$ for $i \in [w]$. This concludes the proof. □

3.2 Construction

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

$$\mathbf{A} = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \\ 1 & 1 \end{pmatrix}, \;\; \mathbf{A}^* = \begin{pmatrix} \frac{1}{a_1} & 0 \\ 0 & \frac{1}{a_2} \\ 0 & 0 \end{pmatrix}, \;\; \mathbf{a}_1^* = \begin{pmatrix} -\frac{1}{a_1} \\ -\frac{1}{a_2} \\ 1 \end{pmatrix},$$

where $a_1, a_2 \leftarrow \mathbb{Z}_p$. 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. Note that we can instantiate H from a hash function $H' : \{0,1\}^* \to G_1$ by generating each output group element of H with H'. More precisely, each output group element of $H(i)$ is defined by $H'(i||\$||j)$, where \$ is a special symbol and $j \in [2k(k+1)]$ specifies the location of the matrices. The symbol \$ can be expressed by encoding, e.g., $0 \to 00$, $1 \to 11$, and $\$ \to 01$. Our scheme for R_{KP} is described as follows.

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

$$\begin{aligned} &\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \;\; \mathbf{A} \leftarrow \mathcal{D}_k, \;\; \mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \;\; \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}, \;\; K \leftarrow \mathcal{K}_\lambda, \\ &\mathsf{pk} := (\mathbb{G}, [\mathbf{A}]_2, [\mathbf{A}^\top \mathbf{k}]_T), \;\; \mathsf{msk} := (\mathbf{A}^*, \mathbf{a}_1^*, \mathbf{B}, \mathbf{k}, K). \end{aligned}$$

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

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

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$**:** It takes pk, msk, and a predicate $y = (\mathbf{y} \in \mathbb{Z}_p^n, f, \psi, t)$ and outputs 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., $d := \max_{i\in[n]} \pi(i)$.

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

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)$**:** It takes pk, ct_x, and sk_y. It computes $b \in \{0,1\}^n$ from x and y as in Definition 2.5. 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.

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

Correctness: For honestly generated ct_x and sk_y such that $R(x,y) = 1$,

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

In the above, we use the relations $\mathbf{A}^\top\mathbf{A}^* = \mathbf{I}_k$ and $\mathbf{A}^\top\mathbf{a}_1^* = \mathbf{0}$. 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, $M' = M$.

3.3 Security

Theorem 3.1. *Let B be the maximum depth of formulae on which $\mathcal{A}$ queries* KeyGen*. Let q_{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*

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

Proof overview. We prove Theorem 3.1 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 [13]. Concretely, we can rewrite $c_{2,i}$ and $k_{2,i}$ in the challenge ciphertext and secret keys as

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

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 $\mathbf{A}\mathbf{s}$ 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 $\mathbf{k} + \mu \mathbf{a}_1^*$ instead of $\mathbf{k}$ for $\mu \leftarrow \mathbb{Z}_p$. To prove each indistinguishability, we utilize the KW technique [23]. In the final hybrid, we can argue that $\mathbf{c}^\top \mathbf{k}$ in the challenge ciphertext is statistically close to a uniform randomness.

Proof. We consider a series of hybrids H_0, H_1, H_2, and $\mathsf{H}_{3,\iota}$ for $i \in \{0, \ldots, q_{\mathsf{sk}}\}$, where 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 H by $\langle \mathcal{A}, \mathsf{H} \rangle_{\mathsf{win}}$, where β is a random bit chosen by the challenger, and β' 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 H_1 as the same as 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

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

$\underline{\mathsf{H}_2}$**.** Next, we define 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 the challenger can access. We denote the first and second elements of $H'(i)$ by $\mathbf{W}_{i,0}$ and $\mathbf{W}_{i,1}$, respectively. In 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 $(\mathbf{W}^\top_{i,0}\mathbf{a}_1, \mathbf{W}^\top_{i,1}\mathbf{a}_1)$. 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 $\overline{\mathbf{A}} = (\mathbf{A} || \mathbf{a}_1)$ 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:

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

In the above, we use the relations $\mathbf{A}^*\mathbf{A}^\top + \mathbf{a}_1^*\mathbf{a}_1^\top = \mathbf{I}_{k+1}$.

$\mathsf{H}_{3,\iota}$. 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 H_2. That is,

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

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,

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

Similarly, a normal secret key is one generated as in H_2. That is,

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

Especially, $\mathbf{k}_1, \ldots, \mathbf{k}_n$ in $k_{2,i}$ is outputs of $\mathsf{Share}(f, \mathbf{k})$. On the other hand, in a semi-functional secret key, $\mathbf{k}_1, \ldots, \mathbf{k}_n$ in $k_{2,i}$ is outputs of $\mathsf{Share}(f, \mathbf{k} + \mu\mathbf{a}_1^*)$ where $\mu \leftarrow \mathbb{Z}_p$. Then, $\mathsf{H}_{3,\iota}$ is the same as H_2 except that the challenge ciphertext and the first ι keys that $\mathcal{A}$ is given are semi-functional.

Lemma 3.2.

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

Proof. To show this, we describe $\mathcal{B}$, which is given an instance of the $\mathcal{D}_k$-MDDH problem $(\mathbb{G}, [\mathbf{A}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$. Let $H' : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times(k+1)} \times \mathbb{Z}_p^{(k+1)\times(k+1)}$ be a random oracle simulated by $\mathcal{B}$ that $\mathcal{A}$ cannot access.

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 $([\mathbf{W}_{i,0}^\top\mathbf{A}]_1, [\mathbf{W}_{i,1}^\top\mathbf{A}]_1)$, 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 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 ct_{x^*} as
$$c_1 = [\mathbf{t}_\beta]_2,\ c_{2,i} = [(x_i\mathbf{W}_{\phi(i),0}^\top + \mathbf{W}_{\phi(i),1}^\top)\mathbf{t}_\beta]_1,\ \ c_3 = e([\mathbf{t}_\beta]_1, [\mathbf{k}]_2)M_\delta.$$
6. $\mathcal{B}$ outputs $\mathrm{true}(\delta = \delta')$, where δ' is an output of $\mathcal{A}$.

The case $\beta = 0$ corresponds to H_2 and the case $\beta = 1$ corresponds to $\mathsf{H}_{3,0}$. □

In the next lemma, we prove the indistinguishability between $\mathsf{H}_{3,\iota-1}$ and $\mathsf{H}_{3,\iota}$. That is, all PPT adversaries cannot distinguish whether the ι-th secret key is normal or semi-functional. To prove this one-secret-key indistinguishability, we introduce core 1-ABE game $\mathsf{G}_\beta^{\text{1-ABE}}$ where $\beta \in \{0,1\}$ such that $\mathsf{G}_0^{\text{1-ABE}}$ and $\mathsf{G}_1^{\text{1-ABE}}$ are computationally indistinguishable. Roughly speaking, the core 1-ABE game is designed so that we can construct a distinguisher between $\mathsf{G}_0^{\text{1-ABE}}$ and $\mathsf{G}_1^{\text{1-ABE}}$ if there exists an adversary that can distinguish $\mathsf{H}_{3,\iota-1}$ and $\mathsf{H}_{3,\iota}$.

It is convenient for us to parametrize the core 1-ABE game by $\eta \in \{1,2\}$ because we also use it in the security proof of our CP-ABE scheme. We use the game with $\eta = 1$ in the security proof of our KP-ABE scheme, and that with $\eta = 2$ in the security proof of our CP-ABE scheme.

Definition 3.1 (Core 1-ABE). *For $\eta \in \{1,2\}$ and $\beta \in \{0,1\}$, we define $\mathsf{G}_{\eta,\beta}^{1\text{-}ABE}$ as Fig 2. In $\mathsf{G}_{\eta,\beta}^{1\text{-}ABE}$, $\mathcal{A}$ can query $\mathcal{O}_X$ and $\mathcal{O}_F$ only once whereas $\mathcal{A}$ can query $\mathcal{O}_R$ polynomially many times. All queries can be done adaptively. Furthermore, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ on which $\mathcal{A}$ queries $\mathcal{O}_X$ and $\mathcal{O}_F$ must satisfy $R(x,y) = 0$. $\mathcal{X}$ and $\mathcal{Y}$ are defined in Definition 2.5. Note that the difference between $\mathsf{G}_{\eta,0}^{1\text{-}ABE}$ and $\mathsf{G}_{\eta,1}^{1\text{-}ABE}$ lies in the input of* Share *in $\mathcal{O}_F$. We define the advantage of $\mathcal{A}$ against $\mathsf{G}_{\eta,\beta}^{1\text{-}ABE}$ as follows:*

$$\mathsf{Adv}_{\mathcal{A},\eta}^{\text{1-ABE}}(\lambda) := |\Pr[\langle \mathcal{A}, \mathsf{G}_{\eta,0}^{\text{1-ABE}} \rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{G}_{\eta,1}^{\text{1-ABE}} \rangle = 1]|.$$

We defer the proof of the indistinguishability between the two games to Section 4.

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

$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mu' \leftarrow \mathbb{Z}_p$, $\mathbf{A} \leftarrow \mathcal{D}_k$, $\overline{\mathbf{B}} \leftarrow \mathbb{Z}_p^{(k+\eta)\times(k+\eta)}$

$\mathbf{d} \leftarrow \mathbb{Z}_p^{k+\eta}$, $\mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+\eta)}$, $L := \emptyset$

$$\mathsf{param} := \begin{cases} (\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{d}, \mathbf{W}) & \eta = 1 \\ (\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{d}, \mathbf{W}, \mathbf{b}_2^*) & \eta = 2 \end{cases}$$

$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_X(\cdot),\mathcal{O}_F(\cdot),\mathcal{O}_R(\cdot)}(\mathsf{param})$

$\underline{\mathcal{O}_X(\cdot)}$

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

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

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

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

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

$\underline{\mathcal{O}_F(\cdot)}$

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

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

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

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

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

$\mathbf{v}_i := \mathbf{B}\mathbf{r}_i$

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

For $i \in [n]$:

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

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

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

  If $t(i) = 1$:

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

  If $t(i) = 0$:

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

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

$\underline{\mathcal{O}_R(\cdot)}$

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

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

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

Output $([\mathbf{W}_{i,0}^\top \mathbf{A}]_1, [\mathbf{W}_{i,1}^\top \mathbf{A}]_1, [\mathbf{W}_{i,0}\mathbf{B}]_1, [\mathbf{W}_{i,1}\mathbf{B}]_1)$

Figure 2: Core 1-ABE game.

Lemma 3.3. *For $\iota \in [q_{\mathsf{sk}}]$, we have*

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

Proof. We consider an adversary $\mathcal{B}$ against $\mathsf{G}^{\text{1-ABE}}_{1,\beta}$ where $\eta = 1$. We describe $\mathcal{B}$'s behavior.

1. $\mathcal{B}$ is given $(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{d}, \mathbf{W})$ from the 1-ABE game.
2. $\mathcal{B}$ sets $\mathbf{k} := \mathbf{W}\mathbf{d}$ and gives $\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_2, [\mathbf{A}^\top \mathbf{k}]_T)$ to $\mathcal{A}$.
3. For query $H(i)$, $\mathcal{B}$ makes a query $\mathcal{O}_R(i)$ and answers with $([\mathbf{W}_{i,0}^\top \mathbf{A}]_1, [\mathbf{W}_{i,1}^\top \mathbf{A}]_1)$.
4. For the challenge query with an attribute x^*, $\mathcal{B}$ flips the coin $\delta \leftarrow \{0,1\}$. Then, $\mathcal{B}$ obtains $(A_0, \{A_i\}_{i\in[m]})$ as the reply of $\mathcal{O}_X(x^*)$. $\mathcal{B}$ returns ct_{x^*} 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 ℓ-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\ell < \iota$ and $y = (\mathbf{y}, f, \psi, t)$, $\mathcal{B}$ computes sk_y as in Eq. (4) by setting $\mathbf{k}_1, \ldots, \mathbf{k}_n \leftarrow \mathsf{Share}(f, \mathbf{k} + \mu \mathbf{a}_1^*)$ with a fresh randomness $\mu \leftarrow \mathbb{Z}_p$.
6. For the ℓ-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 $\mathcal{O}_F(y)$. Then, $\mathcal{B}$ returns sk_y as
$$\mathsf{sk}_y := (P_0, \{P_i\}_{i\in[n]}).$$
7. For the ℓ-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, where $\ell > \iota$ and $y = (\mathbf{y}, f, \psi, t)$, $\mathcal{B}$ computes 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 true$(\delta = \delta')$, where δ' is an output of $\mathcal{A}$.

From Lemma 3.1, the term $\mathbf{k}_i + \sigma_i \mathbf{a}_1^*$ in the reply of $\mathcal{O}_F$ is identically distributed with the i-th output of $\mathsf{Share}(\mathbf{k} + \beta\mu\mathbf{a}_1^*)$. Thus, if the oracles are those in $\mathsf{G}^{\text{1-ABE}}_{1,0}$, $\mathcal{A}$'s view corresponds to $\mathsf{H}_{3,\iota-1}$, and otherwise, it corresponds to $\mathsf{H}_{3,\iota}$. □

Lemma 3.4.

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

Proof. Because $(\mathbf{A}^* || \mathbf{a}_1^*)$ forms a basis, redefining $\mathbf{k}$ as $\mathbf{k} := \mathbf{A}^*\mathbf{z} + z\mathbf{a}_1^*$ 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, $\mathsf{Share}(f, \mathbf{k} + \mu\mathbf{a}_1^*)$ in sk_y, and $\mathbf{c}^\top\mathbf{k}$ in ct_{x^*}. However, $\mathbf{A}^\top\mathbf{k}$ does not contain the information on z because $\mathbf{A}^\top\mathbf{a}_1^* = \mathbf{0}$. Similarly, each $\mathbf{k} + \mu\mathbf{a}_1^*$ also does not contain the information on z because it is masked by fresh randomness μ. Thus, $z\mathbf{c}^\top\mathbf{a}_1^*$ is randomly distributed in $\mathbb{Z}_p$ for $\mathcal{A}$, and so is $\mathbf{c}^\top\mathbf{k}$, unless $\mathbf{c}^\top\mathbf{a}_1^* = 0$. Since $\mathbf{c}$ is randomly chosen from $\mathbb{Z}_p^{k+1}$, $\mathbf{c}^\top\mathbf{a}_1^* = 0$ with a probability $2^{-\Omega(\lambda)}$. If it is not the case, ct_{x^*} does not have information on β, and the lemma holds. □

Thanks to Eq. (1) to (3) and (5) to (7) and Lemma 4.1, Theorem 3.1 holds. □

4 Adaptive Security for Core Component

In this section, we prove the indistinguishability between $\mathsf{G}^{\text{1-ABE}}_{\eta,0}$ and $\mathsf{G}^{\text{1-ABE}}_{\eta,1}$ defined in Definition 3.1. This is formally stated in the following lemma.

Lemma 4.1 (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 $\mathcal{A}$ and $\eta \in \{1,2\}$, there exists a PPT algorithm $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\text{1-ABE}}_{\mathcal{A},\eta}(\lambda) \leq 2^{9B+2}(\mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda) + 2^{-\Omega(\lambda)}).$$

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

Input: $f = (n, w, v, G)$ 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 $\widetilde{\mathsf{Share}}$.

Proof. We prove Lemma 4.1 by extending the KW technique [23]. We omit the variable η from the notation of hybrid games for conciseness, but all hybrids are parametrized by η. 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 $\mathsf{G}_{\mathsf{M0}}^{\text{1-ABE}}$ and $\mathsf{G}_{\mathsf{M1}}^{\text{1-ABE}}$, which satisfy

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

Let $z := (x, y) \in \{0,1\}^R$ on which $\mathcal{A}$ queries $\mathcal{O}_X$ and $\mathcal{O}_F$, respectively. Let $b \in \{0,1\}^n$ be a string computed from z following Definition 2.5. Note that $f(b) = 0$ because the game imposes 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 2.2. 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 $\mathcal{O}_F$ as follows. That is, $\widehat{\mathsf{H}}^{h_\iota}$ is the same as $\widehat{\mathsf{G}}_0^{\text{1-ABE}}$ except that $\mathsf{Share}(f, 0)$ is replaced with $\widetilde{\mathsf{Share}}(f, 0, h_\iota(z))$, which is described in Fig 3. $\mathsf{G}_{\mathsf{M0}}^{\text{1-ABE}}$ is the same as H^{h_L}, and $\mathsf{G}_{\mathsf{M1}}^{\text{1-ABE}}$ is the same as $\mathsf{G}_{\mathsf{M0}}^{\text{1-ABE}}$ except that $\widetilde{\mathsf{Share}}(f, 0, h_L(z))$ is replaced with $\widetilde{\mathsf{Share}}(f, \mu, h_L(z))$.

We prove that

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

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.

$\underline{\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}} = \mathsf{G}^{\text{1-ABE}}_{\mathsf{M1}}}$. 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. 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 σ_{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.

$\underline{\mathsf{G}^{\text{1-ABE}}_{0} \approx_c \mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}}$. Following Lemma 2.1, we prove the two properties:

1. $\mathsf{G}^{\text{1-ABE}}_{0} = \mathsf{H}^{h_0}$ and $\mathsf{H}^{h_L} = \mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}$,
2. $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1} \approx_c \widehat{\mathsf{H}}^{h_\iota}_{0}$ for $\iota \in [L]$.

where $\widehat{\mathsf{H}}^{h_i}_{\beta}$ for $\beta \in \{0,1\}$ is defined in Section 2.4. For item 1, the latter holds because we defined $\mathsf{G}^{\text{1-ABE}}_{\mathsf{M0}}$ 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 $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1} \approx_c \widehat{\mathsf{H}}^{h_\iota}_{0}$. Formally, we show that, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}^{h_{\iota-1}}_{1} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}^{h_\iota}_{0} \rangle = 1]| \leq 2\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda) + 2^{-\Omega(\lambda)}.$$

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

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_ι 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 2.8. Recall that the difference between $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1}$ and $\widehat{\mathsf{H}}^{h_\iota}_{0}$ is the input of $\widetilde{\mathsf{Share}}$. That is, the input is $(f, 0, u_0)$ in $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1}$ and $(f, 0, u_1)$ in $\widehat{\mathsf{H}}^{h_\iota}_{0}$. Thus, in case of $u_0 = u_1$, $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1}$ and $\widehat{\mathsf{H}}^{h_\iota}_{0}$ 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 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 $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,1}$ to $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,3}$ are different from $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1}$ only in $\mathcal{O}_F$ as shown in Fig 4. That is, when O is a gate, $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,1}$ to $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,3}$ are the same as $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1}$. When O is an input wire, these hybrids are defined as follows:

- $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,1}$ is the same as $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1}$ except that $\mathbf{v}_{\pi(i^*)} \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1)$,
- $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,2}$ is the same as $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,1}$ except that random value u is added to σ_{i^*},
- $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,3}$ is the same as $\widehat{\mathsf{H}}^{h_{\iota-1}}_{1,2}$ 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 4.2 to 4.5 and observations so far, Lemma 4.1 holds. □

Lemma 4.2. $|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}^{h_{\iota-1}}_{1} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}^{h_{\iota-1}}_{1,1} \rangle = 1]| \leq \mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{bi}}(\lambda)$.

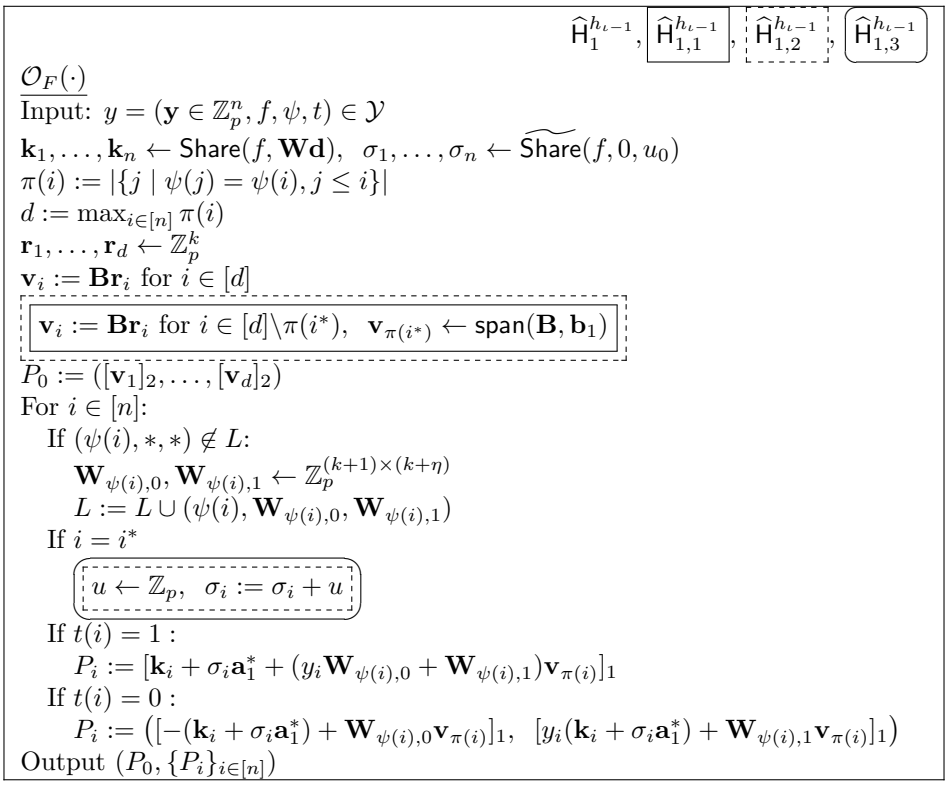


Figure 4: Description of O_F in hybrids.

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 $\mathbf{v}_{\pi(i^*)} \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1)$ in the latter. We show that the $\mathcal{U}_k$-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 $\mathbf{v} \leftarrow \mathbb{Z}_p^{k+1}$. In case of $\eta = 1$, $\mathcal{B}$ just sets $\mathbf{B} := \mathbf{M}$, and then the reduction goes straightforwardly. Thus, we describe the reduction in case of $\eta = 2$. $\mathcal{B}$ chooses $\mathbf{X} \leftarrow \mathsf{GL}_{k+2}(\mathbb{Z}_p)$ and sets

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

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

where $\widehat{\mathbf{M}}$ 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

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

$\mathcal{B}$ 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 $\mathcal{O}_X$ and $\mathcal{O}_R$, $\mathcal{B}$ replies honestly. When $\mathcal{A}$ queries $\mathcal{O}_F$, $\mathcal{B}$ 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

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

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

Proof. We redefine that $\mathbf{W}_{\psi(i^*),b} := \widetilde{\mathbf{W}}_{\psi(i^*),b} + w_{\psi(i^*),b}\mathbf{a}_1^*{\mathbf{b}_1^*}^\top$, where $\widetilde{\mathbf{W}}_{\psi(i^*),b} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+\eta)}$, $w_{\psi(i^*),b} \leftarrow \mathbb{Z}_p$, 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 $\mathcal{O}_R$ because $\mathbf{a}_1^{*\top}\mathbf{A} = \mathbf{0}^\top$ and $\mathbf{b}_1^{*\top}\mathbf{B} = \mathbf{0}^\top$. In $\mathcal{O}_F$, P_i for $i \in \psi^{-1}(\psi(i^*))$ can be written as

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

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 ${\mathbf{b}_1^*}^\top \mathbf{v}_{\pi(i^*)} \neq 0$ with overwhelming probability because $\mathbf{v}_{\pi(i^*)}$ is chosen randomly from $\mathsf{span}(\mathbf{B}, \mathbf{b}_1^*)$.

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

$$\begin{aligned}
&(x\mathbf{W}_{\psi(i^*),0}^\top + \mathbf{W}_{\psi(i^*),1}^\top)\mathbf{c} \\
=&(x\widetilde{\mathbf{W}}_{\psi(i^*),0}^\top + \widetilde{\mathbf{W}}_{\psi(i^*),1}^\top)\mathbf{c} + (xw_{\psi(i^*),0} + w_{\psi(i^*),1})\mathbf{b}_1^*{\mathbf{a}_1^*}^\top \mathbf{c}
\end{aligned}$$

  for some $x \neq y_{i^*}$ from $\mathcal{O}_X$. In both cases, the value $(y_i w_{\psi(i^*),0} + w_{\psi(i^*),1}){\mathbf{b}_1^*}^\top \mathbf{v}_{\pi(i^*)}$ in P_{i^*} is randomly distributed from the viewpoint of $\mathcal{A}$ because this is a pairwise independent function. Thus, adding $u\mathbf{a}_1^*$ to P_{i^*} does not change the entire 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

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

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

Thus, the views of $\mathcal{A}$ in both hybrids are identical unless ${\mathbf{b}_1^*}^\top \mathbf{v}_{\pi(i^*)} = 0$. □

Lemma 4.4. $|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,2}^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}} \rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B},\mathsf{bi}}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}(\lambda)$.

We omit the proof because the proof of this lemma is almost the same as Lemma 4.2.

Lemma 4.5. $\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{1,3}^{h_{\iota-1}} \rangle = 1] = \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_{0}^{h_{\iota}} \rangle = 1]$.

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 σ_{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 σ_a. That is, if O' is a gate g' with a pebble, the values set to its incoming wires are independent of σ_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 σ_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 object O is an input wire i^*. Let a' be an incoming wire of a gate g' that the input wire i^* leads to, that is, $a' = i^*$. In this case, the difference between $\widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\widetilde{\mathsf{Share}}(f, 0, u_1)$ is whether σ_{i^*} is equal 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. □

5 Our CP-ABE Scheme

5.1 Construction

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 $k = 2$.

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 for R_{CP} can be described as follows.

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

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

$\mathsf{Enc}(\mathsf{pk}, x, M)$**:** 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 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., $d := \max_{i\in[n]} \pi(i)$.

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

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

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

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)$**:** It takes pk, ct_x, and sk_y. It computes $b \in \{0,1\}^n$ from x and y as in Definition 2.5. 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.

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

Correctness: For honestly generated ct_x and sk_y such that $R(x, y) = 1$, we have

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

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, $M' = M$.

5.2 Security

Theorem 5.1. *Let B be the maximum depth of a formula used in the challenge ciphertext. Let q_{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*

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

Proof overview. Although the security proof of our CP-ABE scheme also follows the dual system methodology and KW framework [23], it is more complicated than the proof of our KP-ABE scheme. The main reason arises from the fact that we need to use the MDDH assumption as a sort 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 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 sub-group 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 [14, 17]. 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 [13]. 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 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

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

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

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

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 H_0, H_1, H_2, $\mathsf{H}_{3,0}$, $\mathsf{H}_{3,\iota,1}$ to $\mathsf{H}_{3,\iota,3}$ for $\iota \in \{1,\ldots,q_{\mathsf{sk}}\}$, where 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 H by $\langle \mathcal{A}, \mathsf{H}\rangle_{\mathsf{win}}$, where β is a random bit chosen by challenger $\mathcal{C}$, and β' 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 H_1 as the same as 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

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

$\underline{\mathsf{H}_2}$. Next, we define 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 the challenger can access. We denote the first and second elements of $H'(i)$ by $\mathbf{W}_{i,0}$ and $\mathbf{W}_{i,1}$, respectively. In 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;

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

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

$\underline{\mathsf{H}_{3,\iota}}$. 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 H_2. That is,

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

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,

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

For secret keys, we define four secret keys, namely, normal, P-normal, P-SF, and SF. That is, sk_y is defined as

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

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 ι-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 ι-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 ι-th secret key is SF.

Lemma 5.1.

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

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 $(\mathbb{G}, [\mathbf{B}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$. Let $H' : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times(k+2)} \times \mathbb{Z}_p^{(k+1)\times(k+2)}$ be a random oracle simulated by $\mathcal{B}$ that $\mathcal{A}$ cannot access.

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 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 $x^* = (\mathbf{x}, f, \psi, t)$, $\mathcal{B}$ flips the coin $\delta \leftarrow \{0,1\}$ and generates ct_{x^*} as

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

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

Clearly, the case $\beta = 0$ corresponds to 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 5.2. *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 ι-th secret key is normal or P-normal. We describe $\mathcal{B}$, which is given an instance of $\mathcal{D}_k$-MDDH problem, $(\mathbb{G}, [\mathbf{A}]_{1,2}, [\mathbf{t}_\beta]_{1,2})$. Let $H' : \{0,1\}^* \to \mathbb{Z}_p^{(k+1)\times(k+2)} \times \mathbb{Z}_p^{(k+1)\times(k+2)}$ be a random oracle simulated by $\mathcal{B}$ that $\mathcal{A}$ cannot access.

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 τ-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$ such that $\tau < \iota$, $\mathcal{B}$ computes sk_y as
$$k_1 := [\mathbf{As}]_2,\ k_2 := [\mathbf{k} + \mu\mathbf{b}_2^* + \mathbf{W}^\top\mathbf{As}]_1,\ \ k_{3,i} := [(y_i\mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1,$$
where $\mu \leftarrow \mathbb{Z}_p$.
5. For the ι-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes sk_y as
$$k_1 := [\mathbf{t}_\beta]_2,\ k_2 := [\mathbf{k} + \mathbf{W}^\top\mathbf{t}_\beta]_1,\ \ k_{3,i} := [(y_i\mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{t}_\beta]_1.$$
6. For the τ-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$ such that $\tau > \iota$, $\mathcal{B}$ computes sk_y as
$$k_1 := [\mathbf{As}]_2,\ k_2 := [\mathbf{k} + \mathbf{W}^\top\mathbf{As}]_1,\ \ k_{3,i} := [(y_i\mathbf{W}^\top_{\phi(i),0} + \mathbf{W}^\top_{\phi(i),1})\mathbf{As}]_1.$$
7. For the challenge query, $\mathcal{B}$ flips the coin $\delta \leftarrow \{0,1\}$ and generates ct_{x^*} for M_δ as in Eq. (11).
8. $\mathcal{B}$ outputs true($\delta = \delta'$), where δ' 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 $\mathsf{H}_{3,\iota,1}$. □

Lemma 5.3. *For* $\iota \in [q_{\mathsf{sk}}]$, *we have*

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

Proof. We consider an adversary $\mathcal{B}$ against $\mathsf{G}^{\text{1-ABE}}_{2,\beta}$ where $\eta = 2$, which is defined in Definition 3.1. In contrast to the proof of our KP-ABE, $\mathcal{B}$ uses $\mathcal{O}_X$ to generate the ι-th secret key and $\mathcal{O}_F$ to generate the challenge ciphertext. Thus, the reduction seems to prove indistinguishability of two types of ciphertexts. However, we show that this indistinguishability is equivalent to that between the cases where the ι-th secret key is P-normal and P-SF1. We describe $\mathcal{B}$ in the following.

1. $\mathcal{B}$ is given $(\mathbb{G}, \mathbf{A}, [\mathbf{B}]_{1,2}, \mathbf{d}, \mathbf{W}, \mathbf{b}_2^*)$ from the 1-ABE game.
2. $\mathcal{B}$ generates $\mathbf{k} \leftarrow \mathbb{Z}_p^{k+2}$ and gives $\mathsf{pk} = (\mathbb{G}, [\mathbf{B}]_2, [\mathbf{WB}]_1, e([\mathbf{B}]_1, [\mathbf{k}]_2))$ to $\mathcal{A}$.
3. For a query $H(i)$, $\mathcal{B}$ makes a query $\mathcal{O}_R(i)$ and answers with $([\mathbf{W}_{i,0}\mathbf{B}]_1, [\mathbf{W}_{i,1}\mathbf{B}]_1)$.
4. For the challenge query with an attribute $x^* = (\mathbf{x} \in \mathbb{Z}_p^n, f, \psi, t)$, $\mathcal{B}$ flips the coin $\delta \leftarrow \{0,1\}$. Then, $\mathcal{B}$ obtains $(P_0, \{P_i\}_{i\in[n]})$ as the reply of $\mathcal{O}_F(x^*)$. $\mathcal{B}$ returns 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 τ-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$ where $\tau < i$ and $y = (\mathbf{y}, \phi)$, $\mathcal{B}$ computes sk_y as an SF secret key in Eq. (12) with a fresh randomness $\mu \leftarrow \mathbb{Z}_p$.

6. For the τ-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 $\mathcal{O}_X(y)$. Then, $\mathcal{B}$ returns 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 τ-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$ where $\tau > i$ and $y = (\mathbf{y}, \phi)$, $\mathcal{B}$ computes sk_y as a normal secret key in Eq. (12).

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

Then, we implicitly define that $\mathbf{W} := \widetilde{\mathbf{W}} - \frac{\beta\mu' \mathbf{a}_1^* \mathbf{b}_2^{*\top}}{\mathbf{b}_2^{*\top}\mathbf{d}}$ 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

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

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

Next, we look at the distribution of ct_{x^*}. We define that $\mathbf{w}_i := \mathbf{k}_i$. 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 $\mathbf{w}_1, \ldots, \mathbf{w}_n \leftarrow \mathsf{Share}(f, \mathbf{W}\mathbf{d} + \beta\mu'\mathbf{a}_1^*) = \mathsf{Share}(f, \widetilde{\mathbf{W}}\mathbf{d})$. In the above, we use the relations $\mathbf{a}_1^{*\top}\mathbf{A} = \mathbf{0}^\top$ 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 $\mu := -\frac{\mu' \mathbf{a}_1^{*\top} \mathbf{c}}{\mathbf{b}_2^{*\top}\mathbf{d}}$. Note that μ' appear only in k_2 in the ι-th secret key. Thus, μ is randomly distributed in $\mathbb{Z}_p$. This concludes the proof. □

Lemma 5.4. *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 5.2.

Lemma 5.5.

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

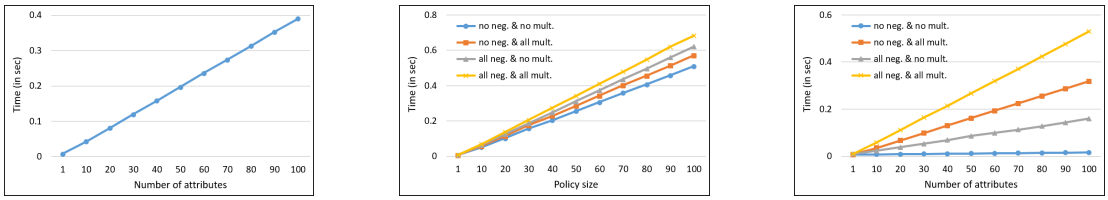
Table 2: Specifications of devices for our benchmarks.

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
iPhone XR	iOS 12.2	Apple A12 Bionic	Apple LLVM 10.0.1
Pixel 3	Android 9	Qualcomm Snapdragon 845	Android clang 8.0.2

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 pk, $\mathbf{k} + \mu\mathbf{b}_2^*$ in sk_y, and $\mathbf{d}^\top\mathbf{k}$ in 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 μ. 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, ct_{x^*} does not have information on β, and the lemma holds. □

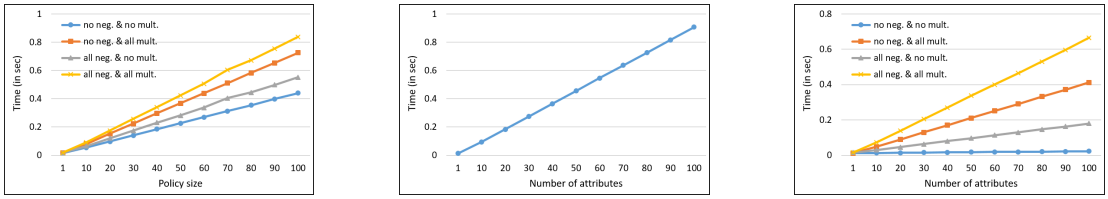
Thanks to Eq. (8) to (10) and (13) to (17) and Lemma 4.1, Theorem 5.1 holds. □

6 Implementation and Evaluation



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

Figure 5: Benchmarks of our KP-ABE on PC.

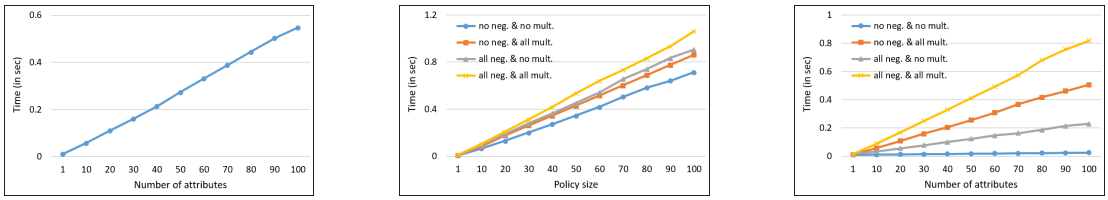


(a) Enc (b) KeyGen (c) Dec

Figure 6: Benchmarks of our CP-ABE on PC.

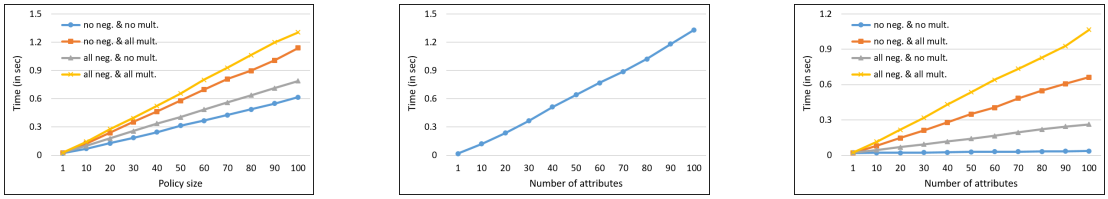
We implement our KP-ABE and CP-ABE schemes and measure the benchmarks of our schemes on an ordinary personal computer (PC) and two smartphones, iPhone XR and Pixel 3. Their specifications are shown in Table 2. Theoretical comparisons with previous schemes are presented in Section 7.

We 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



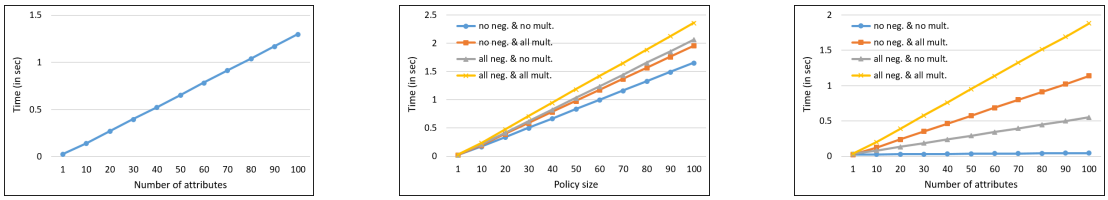
(a) Enc (b) KeyGen (c) Dec

Figure 7: Benchmarks of our KP-ABE on iPhone XR.



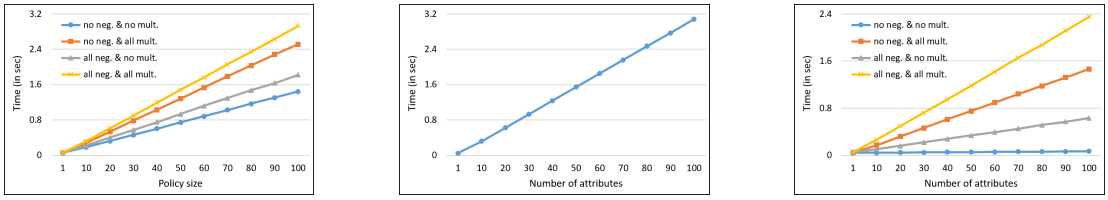
(a) Enc (b) KeyGen (c) Dec

Figure 8: Benchmarks of our CP-ABE on iPhone XR.



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

Figure 9: Benchmarks of our KP-ABE on Pixel 3.



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

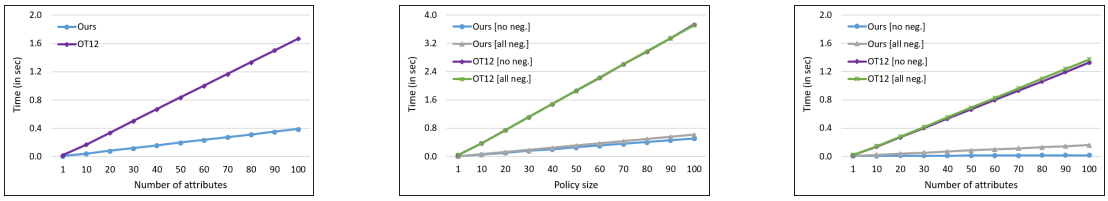
Figure 10: Benchmarks of our CP-ABE on Pixel 3.

logarithm problem in a finite field [21, 22]. The running times of time-consuming operations on the PC are listed in Table 3.

As we can see in Sections 3.2 and 5.1, 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.

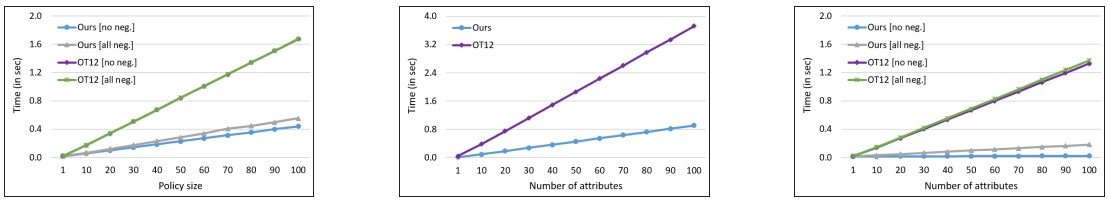
To clarify the effects of these factors, we consider the four types of formulae.

1. no negations and multi-uses (no neg. & no mult.):
   i.e., (Label-1:v_1 AND Label-2:v_2 AND ...),



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

Figure 11: Comparison of KP-ABE between ours and OT12 on PC.



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

Figure 12: Comparison of CP-ABE between ours and OT12 on PC.

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

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 5 and 6, iPhone XR in Fig 7 and 8, and Pixel 3 in Fig 9 and 10. 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 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 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 since the number of negation affects that of exponentiation in G_1 while the number of multi-use affects that of heavier pairings.

Comparison with OT12. We also implement KP and CP schemes by Okamoto and Takashima in [29] (OT12), which are the only schemes that support OT negation and unboundedness, and thus whose functionalities are the closest to our schemes among known ABE schemes. The comparison between our schemes and OT12 on PC is presented in Fig 11 and 12, which shows that our schemes achieve significant speedups in every algorithm. We compare them in the one-use restriction of labels (no multi-use), which corresponds to item 1 and item 2 in the four cases, since OT12 does not support multi-use of labels. Hence, the blue and gray lines in Fig 5 are the same as those in Fig 11 up to scale (similarly in Fig 6 and Fig 12). In contrast to our schemes, negation hardly affects the efficiency in OT12. Note that although we can utilize a bounded number of multi-use of labels by preparing multiple nominal labels for each single label in OT12, this significantly affects the efficiency. For example, when we set the bound as 10, this slows down Enc in KP-ABE or KeyGen in CP-ABE by 10 times.

CCA security. In practice, the chosen ciphertext attack (CCA) security is a de facto standard and desirable security requirement. Fujisaki-Okamoto conversion [16] is not suitable for our case because it requires the decryption algorithm to run the encryption algorithm, which causes a significant efficiency loss. However, our schemes can be efficiently converted to CCA secure ones via Boneh-Katz conversion [11] in a similar manner to [28].

7 Theoretical Comparison

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	-

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

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

and Chase [1], the basic scheme of OT12 by Okamoto and Takashima [29], and the asymmetric variant of GPSW06 by Goyal et al. [18] (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 3.2 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:

- 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 ($I = I_t + I_f$).

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	-

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] Agrawal, S., Chase, M.: FAME: Fast attribute-based message encryption. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 665–682. ACM Press (Oct / Nov 2017)

[2] Agrawal, S., Chase, M.: Simplifying design and analysis of complex predicate encryption schemes. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 627–656. Springer, Heidelberg (Apr / May 2017)

[3] Attrapadung, N.: Unbounded dynamic predicate compositions in attribute-based encryption. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 34–67. Springer, Heidelberg (May 2019)

[4] Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (Mar 2011)

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

[6] Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Santis, A.D. (ed.) EUROCRYPT'94. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (May 1995)

[7] Bellare, M., Rogaway, P.: The exact security of digital signatures: How to sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (May 1996)

[8] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy. pp. 321–334. IEEE Computer Society Press (May 2007)

[9] Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (Aug 2004)

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

[11] Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (Feb 2005)

[12] Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51(4), 557–594 (2004)

[13] Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (Apr 2015)

[14] Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded ABE via bilinear entropy expansion, revisited. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 503–534. Springer, Heidelberg (Apr / May 2018)

[15] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for Diffie-Hellman assumptions. Journal of Cryptology 30(1), 242–288 (Jan 2017)

[16] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)

[17] Gong, J., Dong, X., Chen, J., Cao, Z.: Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 624–654. Springer, Heidelberg (Dec 2016)

[18] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 89–98. ACM Press (Oct / Nov 2006), available as Cryptology ePrint Archive Report 2006/309

[19] Jafargholi, Z., Kamath, C., Klein, K., Komargodski, I., Pietrzak, K., Wichs, D.: Be adaptive, avoid overcommitting. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 133–163. Springer, Heidelberg (Aug 2017)

[20] Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. Journal of Cryptology 26(2), 191–224 (Apr 2013)

[21] Kim, T., Barbulescu, R.: Extended tower number field sieve: A new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 543–571. Springer, Heidelberg (Aug 2016)

[22] Kim, T., Jeong, J.: Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. In: Fehr, S. (ed.) PKC 2017, Part I. LNCS, vol. 10174, pp. 388–408. Springer, Heidelberg (Mar 2017)

[23] Kowalczyk, L., Wee, H.: Compact adaptively secure ABE for NC^1 from k-lin. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 3–33. Springer, Heidelberg (May 2019)

[24] Lewko, A., Waters, B.: Decentralizing attribute-based encryption. Cryptology ePrint Archive, Report 2010/351 (2010), `http://eprint.iacr.org/2010/351`

[25] Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (May / Jun 2010)

[26] Lewko, A.B., Sahai, A., Waters, B.: Revocation systems with very small private keys. In: 2010 IEEE Symposium on Security and Privacy. pp. 273–285. IEEE Computer Society Press (May 2010)

[27] Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (Feb 2010)

[28] Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (Aug 2010)

[29] Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (Dec 2012)

[30] Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Ning, P., De Capitani di Vimercati, S., Syverson, P.F. (eds.) ACM CCS 2007. pp. 195–203. ACM Press (Oct 2007)

[31] Sahai, A., Waters, B.R.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (May 2005)

[32] Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (Aug 2009)

[33] Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (Feb 2014)

[34] Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: A framework and compact constructions for non-monotonic attribute-based encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 275–292. Springer, Heidelberg (Mar 2014)