Duality in ABE: Converting Attribute Based Encryption for Dual Predicate and Dual Policy via Computational Encodings

Nuttapong Attrapadung
AIST, Japan
n.attrapadung@aist.go.jp

Shota Yamada
AIST, Japan
yamada-shota@aist.go.jp

Abstract

We show a generic conversion that converts an attribute based encryption (ABE) scheme for arbitrary predicate into an ABE scheme for its *dual predicate*. In particular, it can convert key-policy ABE (KP-ABE) into ciphertext-policy ABE (CP-ABE), and vice versa, for dually related predicates. It is generic in the sense that it can be applied to arbitrary predicates. On the other hand, it works only within the generic ABE framework recently proposed by Attrapadung (Eurocrypt'14), which provides a generic compiler that compiles a simple primitive called *pair encodings* into fully secure ABE. Inside this framework, Attrapadung proposed the first generic dual conversion that works only for subclass of encodings, namely, *perfectly secure encodings*. However, there are many predicates for which realizations of such encodings are not known, and hence the problems of constructing fully secure ABE for their dual predicates were left unsolved.

In this paper, we revisit the dual conversion of Attrapadung, and show that, somewhat surprisingly, the very same conversion indeed also works for broader classes of encodings, namely, *computationally secure encodings*. Consequently, we thus solve the above open problems as we obtain the first fully secure realizations of completely-unbounded CP-ABE and CP-ABE with short keys for Boolean formulae, via applying the conversion to previously proposed KP-ABE.

Moreover, we provide a generic conversion that converts ABE into its *dual-policy* variant. Dual-policy ABE (DP-ABE) conjunctively combines both KP-ABE and CP-ABE into one primitive, and hence can be useful in general-purpose applications. As for instantiations, we obtain the first realizations of fully secure DP-ABE for formulae, unbounded DP-ABE for formulae, and DP-ABE for regular languages. The latter two systems are the first to realize such functionalities, let alone are fully secure.

Keywords. Attribute-based encryption, Dual scheme conversion, Key-policy, Ciphertext-policy, Dual-policy, Full security, Dual system encryption, Pair Encoding, Functional Encryption.

An extended abstract of this paper appears in the proceedings of CT-RSA 2015. This is the full version.

1 Introduction

Attribute-based encryption (ABE), introduced by Sahai and Waters [32], is a useful paradigm that generalizes traditional public key encryption. Instead of encrypting to a target recipient, a sender can specify in a more general way about who should be able to view the message. In ABE for *predicate* R, which is a boolean function $R : \mathbb{X} \times \mathbb{Y} \to \{0, 1\}$, a private key, which is issued by an authority, is associated with an attribute $X \in \mathbb{X}$, while a ciphertext encrypting a message M is associated with an attribute $Y \in \mathbb{Y}$. A key for X can decrypt a ciphertext for Y if and only if $R(X, Y) = 1$. In a *key-policy* type of ABE (KP-ABE) [21], any $X \in \mathbb{X}$ is viewed as a *policy* function $X : \mathbb{Y} \to \{0, 1\}$ and the predicate evaluation is defined as $R(X, Y) = X(Y)$. On the other hand, in a *ciphertext-policy* type of ABE (CP-ABE) [7], any $Y \in \mathbb{Y}$ is viewed as a function $Y : \mathbb{X} \to \{0, 1\}$ where we define $R(X, Y) = Y(X)$. Perhaps, the most well-known ABE is for *Boolean formulae predicate*, considered by Goyal *et al.* [21], where policy functions are Boolean formulae over attributes and inputs to functions are Boolean assignments of attributes.

Duality in ABE. In this paper, we study the *duality* in ABE. For a predicate $R : \mathbb{X} \times \mathbb{Y} \to \{0, 1\}$, we define its *dual predicate* $\bar{R} : \mathbb{Y} \times \mathbb{X} \to \{0, 1\}$ as

$$\bar{R}(Y, X) = R(X, Y).$$

Hence, key-policy and ciphertext-policy ABE are dual to each other in the sense that, when we view X as a function, ABE for R is of key-policy type, while its dual, ABE for $\bar{R}$, is of ciphertext-policy type.

Although any predicate and its dual are related by a very simple definition, ABE systems for both predicates are usually constructed separately and their security proofs are obtained using different techniques. In fact, until only recently, there was no known *generic* method that converts ABE into its dual. A first attempt for conversion was early done by Goyal *et al.* [22] but for only *specific* predicates, namely, they showed how to convert any KP-ABE into CP-ABE for bounded-size Boolean formulae. Only recently, Attrapadung [2] proposed the first *generic* dual conversion. It is generic in the sense that it can be applied to *arbitrary predicate*. More precisely, in [2], a generic framework for constructing fully-secure ABE was proposed, and inside the framework, a dual conversion was introduced. We first briefly describe the framework of [2].

Framework and Generic Dual Conversion of [2]. The framework of [2] provides an abstraction of the *dual system encryption* approaches, introduced by Waters [34] and extended by many works [25, 28, 29, 27, 36]. The framework of [2] decouples what seem to be an essential underlying primitive in the dual system approaches, called *pair encoding* schemes for predicates, and provides a generic construction that compiles any secure pair encoding for a predicate in consideration to a fully secure ABE for that predicate.[1] The security of encodings comes in two flavors: an *information-theoretical* notion, which captures the traditional dual system approach, and a *computational* notion, which generalizes the techniques in the ABE of Lewko and Waters [27]. Both notions imply fully secure ABE. However, it is the computational notion that empowers the framework of [2], since by using this notion, the first

[1] In our paper, we use "attribute based encryption" to refer to *public-index predicate encryption*, which is a sub-class of functional encryption categorized in [10]. In [2], the same class was referred as "functional encryption" (FE).

fully secure schemes are obtained in [2] for many ABE primitives of which only selectively secure constructions were known before, including KP-ABE for regular languages [35], KP-ABE for Boolean formulae with constant-size ciphertexts [6], and (completely) unbounded KP-ABE for Boolean formulae [26, 31]. In fact, the latter two predicates are special cases of a new predicate called *key-policy over doubly spatial encryption* (KP-DSE), introduced and constructed in [2]. In addition, only the dual of the first ABE above, namely, CP-ABE for regular languages was also directly constructed in [2].

The first generic dual conversion was then given in [2]. It works by converting any pair encoding for R into a pair encoding for its dual, $\bar{R}$. A fully secure ABE scheme for $\bar{R}$ is then obtained via the generic construction. However, in [2], only the case for information-theoretical security of encodings was proved to be preserved via the conversion. Hence, it is not applicable to computationally secure encodings, which empower the framework of [2] in the first place. In particular, fully secure realizations of CP-ABE primitives that are the duals of KP-ABE for the predicates above, namely, unbounded CP-ABE for formulae and constant-size CP-ABE for formulae, and their generalization, ciphertext-policy over DSE (CP-DSE), have been left as open problems. To this end, our first goal is to provide a generic dual conversion that preserves computational security of encodings.

Dual-Policy ABE. Key-policy and ciphertext-policy types are useful in different applications. KP-ABE specifies policies over data attributes, and hence is useful for content-based access control. CP-ABE specifies policies over receiver attributes, and hence is useful for access control that directly specifies receiver policies. In order to make the most advantages of both types, a combined type called *dual-policy* ABE (DP-ABE) was proposed in [5]. DP-ABE conjunctively combines two predicates, namely, a predicate R and its dual predicate $\bar{R}$. The dual-policy predicate, denoted $[R \wedge \bar{R}] : (\mathbb{X} \times \mathbb{Y}) \times (\mathbb{Y} \times \mathbb{X}) \to \{0, 1\}$, is defined by

$$[R \wedge \bar{R}]((X, Y'), (Y, X')) = R(X, Y) \wedge \bar{R}(Y', X').$$

DP-ABE is already found useful in real-world applications due to its flexibility [1]. However, there was no known generic method to combine an ABE and its dual to obtain DP-ABE. It is our second goal to construct a generic conversion that converts ABE into its dual-policy variant.

Our Contributions. We revisit the generic dual conversion of [2] and prove that, somewhat surprisingly, the very same conversion indeed preserves the computational security of encodings. Hence, by applying it to the KP-DSE of [2], we immediately obtain the first fully secure CP-DSE. This implies the first fully secure realizations of completely unbounded CP-ABE for formulae and constant-size-*key* CP-ABE for formulae. We note that constant-size ciphertexts in KP-ABE (of [2]) becomes constant-size keys due to the duality.

We achieve the new theorem of the conversion by a very simple proof that relies on two ingredients. First, we restrict the syntax of pair encodings to the class we call *normal* pair encodings by posing a new simple requirement. Nevertheless, this restriction seems natural and does not affect any concrete pair encoding schemes proposed so far in [2]. Second, we *relax* the computational security but in such a way that the generic ABE construction still compiles encodings to fully secure ABE. Moreover, since we relax the security, all existing computationally secure encodings will satisfy the relaxed notion. The only drawback is that the reduction cost for the resulting ABE will not have the same tightness as in the framework of [2], which achieve $O(q_1)$ reduction to the underlying assumption. The converted ABE, however, achieves $O(q_{\text{all}})$ reduction cost, where q_1, q_{all} are the number of pre-challenge queries and all queries, respectively. To this end, we also directly construct a new CP-DSE scheme and prove its security with tightness $O(q_1)$.

We then propose a generic method to conjunctively combine any two pair encoding schemes. Hence, by combining with the generic dual conversion above, we obtain a generic conversion that converts any normal pair encoding scheme for R into a pair encoding for its *dual-policy*, namely, $[R \wedge \bar{R}]$. This implies the first realizations of fully secure DP-ABE for formulae, DP-DSE, unbounded DP-ABE for formulae, and DP-ABE for regular languages. The latter three systems are the first to realize such primitives, let alone are fully secure.

1.1 Our Approach

Recapturing the Framework of [2]. In the generic construction of ABE for R of [2], a ciphertext CT encrypting M, and a key SK take the forms of

$$\mathsf{CT} = (\boldsymbol{C}, C_0) = (g_1^{\boldsymbol{c}_Y(\boldsymbol{s},\boldsymbol{h})},\ Me(g_1, g_1)^{\alpha s_0}), \qquad \mathsf{SK} = g_1^{\boldsymbol{k}_X(\alpha,\boldsymbol{r},\boldsymbol{h})}$$

where $\boldsymbol{c}_Y$ and $\boldsymbol{k}_X$ are *encodings* of attributes Y and X associated to a ciphertext and a key, respectively. Here, g_1 is a generator of subgroup of order p_1 of $\mathbb{G}$, which is a symmetric bilinear group of composite order $N = p_1p_2p_3$ with bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The bold fonts denote vectors. Intuitively, α plays the role of a master key, $\boldsymbol{h}$ represents common variables (or called parameters). These define a public key $\mathsf{PK} = (g_1^{\boldsymbol{h}}, e(g_1, g_1)^{\alpha})$. $\boldsymbol{s}, \boldsymbol{r}$ represents randomness in the ciphertext and the key, respectively, with s_0 being the first element in $\boldsymbol{s}$. The pair $(\boldsymbol{c}_Y, \boldsymbol{k}_X)$ form a *pair encoding* scheme for predicate R. It is exactly this primitive on which the framework of [2] studied and gave sufficient conditions for correctness (when $R(X, Y) = 1$) and security (when $R(X, Y) = 0$) so that, roughly speaking, the ABE scheme defined with CT, SK as above would be correct and fully secure (see more detail in in §3). We refer the intuition for defining the computational security of encodings to [2], but informally recapture it here. The security requires that for $R(X, Y) = 0$, the following two distributions are computationally indistinguishable:

$$\left(g_2^{\boldsymbol{c}_Y(\boldsymbol{s},\boldsymbol{h})},\ g_2^{\boldsymbol{k}_X(0,\boldsymbol{r},\boldsymbol{h})}\right) \qquad \text{and} \qquad \left(g_2^{\boldsymbol{c}_Y(\boldsymbol{s},\boldsymbol{h})},\ g_2^{\boldsymbol{k}_X(\alpha,\boldsymbol{r},\boldsymbol{h})}\right),$$

where Y, X are chosen by the adversary. It has two sub-notions. For the notion where Y is queried before X, it is called *selective* master-key hiding. On the other hand, if X is queried before Y, we call *co-selective* master-key hiding. The naming mimics the (co-)selective security of ABE. These elements are defined over g_2, a generator of p_2-order subgroup of $\mathbb{G}$, and are only used in the proof.

The main idea for the generic dual conversion of [2] is natural: *simply using key encodings to define ciphertext encodings in the dual predicate, and vice versa.* More precisely, from a pair encoding $(\boldsymbol{c}_Y, \boldsymbol{k}_X)$ for R, a pair encoding $(\bar{\boldsymbol{c}}_X, \bar{\boldsymbol{k}}_Y)$ for $\bar{R}$ is constructed as

$$\bar{\boldsymbol{k}}_Y(\bar{\alpha}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{h}}) := \big(\boldsymbol{c}_Y(\boldsymbol{s}, \boldsymbol{h}),\ \bar{\alpha} + \bar{\phi}s_0\big), \qquad \bar{\boldsymbol{c}}_X(\bar{\boldsymbol{s}}, \bar{\boldsymbol{h}}) := \big(\boldsymbol{k}_X(\bar{\phi}\bar{s}_0, \boldsymbol{r}, \boldsymbol{h}),\ \bar{s}_0\big),$$

where $\bar{\boldsymbol{h}} := (\boldsymbol{h}, \bar{\phi})$, $\bar{\boldsymbol{r}} := \boldsymbol{s}$, $\bar{\boldsymbol{s}} := (\bar{s}_0, \boldsymbol{r})$. We leave the explanation to §4 and only motivate here for non-triviality of proving the preservability of the computational security through the conversion.

Difficulty and Our Idea. The security of original encodings only provides the indistinguishability of $\boldsymbol{k}$ for the case of $\alpha = 0$ and α is random. To establish the reduction, we need to use it to prove the indistinguishability of $\bar{\boldsymbol{k}}$ for the case of $\bar{\alpha} = 0$ and $\bar{\alpha}$ is random. However, the non-triviality here stems from the fact that $\bar{\boldsymbol{k}}$ is defined from $\boldsymbol{c}$, where we do not have a sort of indistinguishability in the first place! We resolve this using a simple technique that establishes the "link" from $\boldsymbol{k}$ to $\bar{\boldsymbol{k}}$ via simulation of the variable $\bar{\phi}$. Without going into details here, in order to do so, we only additionally require s_0 to be given out in $\boldsymbol{c}_Y(\boldsymbol{s}, \boldsymbol{h})$. But this restriction is natural and is satisfied by all the pair encodings proposed so far [2]. We thus call it the *normality* of pair encodings.

Our theorems state that if the original encoding is *selectively* master-key hiding, then the converted encoding for the dual is *co-selectively* master-key hiding, and vice versa. This follows intuitively from the fact that we swap key encodings with ciphertext encodings, and hence the order of queries from the adversary is also swapped. There is a caveat that while the original *selective* notion of [2] allows polynomially many key encoding queries, which results in tighter reduction for ABE, our conversion can deal with only one query. In other words, we relax the selective notion so that it will be preserved via the conversion. Nevertheless, this will affect only the reduction tightness of the resulting ABE, where the reduction will become $O(q_{\mathrm{all}})$, instead of $O(q_1)$ as in [2].

Other Related Work. In this work, we allow only efficient tools, namely, bilinear groups. When basing on stronger (but much less efficient) tools, such as multi-linear maps [14, 12], or cryptographic obfuscations [16], we can obtain ABE and FE for very general classes of predicates such as poly-size circuits [15, 20], or Turing machines [18, 19]. For these general classes, there were no known generic dual conversion. For the circuit predicate, KP-ABE can be converted into CP-ABE but for only *bounded-size circuits*, using universal circuits [16]. We remark that, until recently, all known ABE systems for these general classes are only selectively secure (or fully secure but with exponential reductions). Fully secure KP-ABE systems for circuits are recently proposed in [17, 3]. The first (fully secure) CP-ABE for unbounded-size circuits was proposed also in [3].

2 Preliminaries

Predicate Family. We consider a predicate family $R = \{R_\kappa\}_{\kappa \in \mathbb{N}^c}$, for some constant $c \in \mathbb{N}$, where a relation $R_\kappa : \mathbb{X}_\kappa \times \mathbb{Y}_\kappa \to \{0,1\}$ is a predicate function that maps a pair of key attribute in a space $\mathbb{X}_\kappa$ and ciphertext attribute in a space $\mathbb{Y}_\kappa$ to $\{0,1\}$. The family index $\kappa = (n_1, n_2, \ldots)$ specifies the description of a predicate from the family. We mandate the first entry n_1 in κ to specify the arithmetic domain, *e.g.,* in composite-order setting, it is $\mathbb{Z}_N$ (*i.e.,* $n_1 = N$).

Dual Predicate. For a predicate $R : \mathbb{X} \times \mathbb{Y} \to \{0,1\}$, its dual predicate is defined by $\bar{R} : \bar{\mathbb{X}} \times \bar{\mathbb{Y}} \to \{0,1\}$ where $\bar{\mathbb{X}} = \mathbb{Y}, \bar{\mathbb{Y}} = \mathbb{X}$ and $\bar{R}(X, Y) := R(Y, X)$.

ABE Syntax. An ABE scheme for predicate R consists of four algorithms:

- $\mathsf{Setup}(1^\lambda, \kappa) \to (\mathsf{PK}, \mathsf{MSK})$: takes as input a security parameter 1^λ and a family index κ of predicate family R, and outputs a master public key PK and a master secret key MSK.
- $\mathsf{Encrypt}(Y, M, \mathsf{PK}) \to \mathsf{CT}$: takes as input a ciphertext attribute $Y \in \mathbb{Y}_\kappa$, a message $M \in \mathcal{M}$, and public key PK. It outputs a ciphertext CT.

- $\mathsf{KeyGen}(X, \mathsf{MSK}, \mathsf{PK}) \rightarrow \mathsf{SK}$: takes as input a key attribute $X \in \mathbb{X}_\kappa$ and the master key MSK. It outputs a secret key SK.
- $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK}) \rightarrow M$: given a ciphertext CT with its attribute Y and the decryption key SK with its attribute X, it outputs a message M or $\perp$.

We refer the (standard) definitions of correctness and security of ABE to [2].

Composite-order Bilinear Groups. We use bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3$, where p_1, p_2, p_3 are distinct primes, with an efficient bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. A bilinear group generator $\mathcal{G}(\lambda)$ takes as input a security parameter λ and outputs $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3)$. Let $\mathbb{G}_{p_i}$ be the subgroup of order p_i of $\mathbb{G}$. We note that, nevertheless, we will not directly use properties of composite-order groups (such as orthogonality, subgroup decision assumptions) here. This is since the framework of [2] essentially decouples pair encoding schemes so that they need not incorporate such properties.

3 Pair Encoding Scheme

We recall the definition of pair encoding schemes as given in [2]. A pair encoding scheme for predicate family R consists of four deterministic algorithms given by $\mathsf{P} = (\mathsf{Param}, \mathsf{Enc1}, \mathsf{Enc2}, \mathsf{Pair})$:

- $\mathsf{Param}(\kappa) \rightarrow n$. It takes as input an index κ and outputs n, which specifies the number of *common variables* in $\mathsf{Enc1}$, $\mathsf{Enc2}$. For default notation, let $\boldsymbol{h} = (h_1, \ldots, h_n)$ denote the common variables.
- $\mathsf{Enc1}(X, N) \rightarrow \big(\boldsymbol{k} = (k_1, \ldots, k_{m_1});\ m_2\big)$. It takes as inputs $X \in \mathbb{X}_\kappa$, $N \in \mathbb{N}$, and outputs a sequence of polynomials $\{k_i\}_{i\in[1,m_1]}$ with coefficients in $\mathbb{Z}_N$, and $m_2 \in \mathbb{N}$. We require that each polynomial k_i is a *linear combination of monomials* $\alpha, r_j, h_k r_j$, where $\alpha, r_1, \ldots, r_{m_2}, h_1, \ldots, h_n$ are variables.
- $\mathsf{Enc2}(Y, N) \rightarrow \big(\boldsymbol{c} = (c_1, \ldots, c_{w_1});\ w_2\big)$. It takes as inputs $Y \in \mathbb{Y}_\kappa$, $N \in \mathbb{N}$, and outputs a sequence of polynomials $\{c_i\}_{i\in[1,w_1]}$ with coefficients in $\mathbb{Z}_N$, and $w_2 \in \mathbb{N}$. We require that each polynomial c_i is a *linear combination of monomials* $s_j, h_k s_j$, where $s_0, s_1, \ldots, s_{w_2}, h_1, \ldots, h_n$ are variables.
- $\mathsf{Pair}(X, Y, N) \rightarrow \boldsymbol{E}$. It takes as inputs X, Y, N, and output $\boldsymbol{E} \in \mathbb{Z}_N^{m_1 \times w_1}$.

Correctness. First, we require that for $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, N)$, $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, N)$, $\boldsymbol{E} \leftarrow \mathsf{Pair}(X, Y, N)$, we have that if $R_N(X, Y) = 1$, then $\boldsymbol{kEc}^\top = \alpha s_0$. We note that since we can write $\boldsymbol{kEc}^\top = \sum_{i\in[1,m_1], j\in[1,w_1]} E_{i,j} k_i c_j$, this correctness amounts to check if there is a linear combination of $k_i c_j$ terms summed up to αs_0. Second, for $p|N$, if we let $\mathsf{Enc1}(X, N) \rightarrow (\boldsymbol{k}; m_2)$ and $\mathsf{Enc1}(X, p) \rightarrow (\boldsymbol{k}'; m_2)$, then $\boldsymbol{k} \bmod p = \boldsymbol{k}'$. The requirement for $\mathsf{Enc2}$ is similar.

Notation. In what follows, we denote $\boldsymbol{h} = (h_1, \ldots, h_n)$, $\boldsymbol{r} = (r_1, \ldots, r_{m_2})$, $\boldsymbol{s} = (s_0, s_1, \ldots, s_{w_2})$. We will often use subscripts and write $\boldsymbol{k}_X$ and $\boldsymbol{c}_Y$ to emphasize the attributes X, Y.

Properties. As identified in [2], every pair encoding scheme straightforwardly satisfies the following two properties symbolically. *Parameter-vanishing* states the identity $\boldsymbol{k}(\alpha, \boldsymbol{0}, \boldsymbol{h}) = \boldsymbol{k}(\alpha, \boldsymbol{0}, \boldsymbol{0})$. *Linearity* states the identities: $\boldsymbol{k}(\alpha_1, \boldsymbol{r}_1, \boldsymbol{h}) + \boldsymbol{k}(\alpha_2, \boldsymbol{r}_2, \boldsymbol{h}) = \boldsymbol{k}(\alpha_1 + \alpha_2, \boldsymbol{r}_1 + \boldsymbol{r}_2, \boldsymbol{h})$

for $\boldsymbol{k}$, and $\boldsymbol{c}(\boldsymbol{s}_1,\boldsymbol{h})+\boldsymbol{c}(\boldsymbol{s}_2,\boldsymbol{h})=\boldsymbol{c}(\boldsymbol{s}_1+\boldsymbol{s}_2,\boldsymbol{h})$ for $\boldsymbol{c}$. Combining the two identities for $\boldsymbol{k}$, we have that

$$\boldsymbol{k}(\alpha_1,\boldsymbol{0},\boldsymbol{0})+\boldsymbol{k}(\alpha_2,\boldsymbol{r},\boldsymbol{h})=\boldsymbol{k}(\alpha_1+\alpha_2,\boldsymbol{r},\boldsymbol{h}) \tag{1}$$

Normal Pair Encoding. Towards proving the security of our dual conversion, we require a new property for pair encoding. We formalize it as *normality*. This restriction is natural and all pair encoding schemes proposed so far [2, 36] are not affected by this.

Definition 1 (Normal Pair Encoding)**.** We call a pair encoding scheme *normal* if s_0 is a polynomial in the sequence $\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})$. Wlog, we denote $c_1=s_0$ (the first polynomial in $\boldsymbol{c}$).

3.1 Computational Security Definitions of Pair Encoding

We use the same computational security notion of pair encoding as defined in [2], albeit we reformalize with additional refinement regarding the number of queries that can be asked by the adversary. The notion consists of two sub-notions: *selectively secure* and *co-selectively secure master-key hiding* ($\mathsf{SMH},\mathsf{CMH}$) in a bilinear group generator $\mathcal{G}$. We first define the following game template, denoted as $\mathsf{Exp}_{\mathcal{G},\mathsf{P},\mathsf{G},b,\mathcal{A},t_1,t_2}(\lambda)$, for pair encoding P, a flavor $\mathsf{G}\in\{\mathsf{CMH},\mathsf{SMH}\}$, $b\in\{0,1\}$, and $t_1,t_2\in\mathbb{N}$. It takes as input the security parameter λ and does the experiment with the adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$, and outputs b'. Denote by st a state information by $\mathcal{A}$. The game is defined as:

$$\begin{aligned}\mathsf{Exp}_{\mathcal{G},\mathsf{P},\mathsf{G},b,\mathcal{A},t_1,t_2}(\lambda): &(\mathbb{G},\mathbb{G}_T,e,N,p_1,p_2,p_3)\leftarrow\mathcal{G}(\lambda),\ g_i\xleftarrow{\$}\mathbb{G}_{p_i}(\text{for } i=1,2,3),\\ &\alpha\xleftarrow{\$}\mathbb{Z}_N,\ n\leftarrow\mathsf{Param}(\kappa),\ \boldsymbol{h}\xleftarrow{\$}\mathbb{Z}_N^n,\\ &\mathsf{st}\leftarrow\mathcal{A}_1^{\mathcal{O}^1_{\mathsf{G},b,\alpha,\boldsymbol{h}}(\cdot)}(g_1,g_2,g_3),\ b'\leftarrow\mathcal{A}_2^{\mathcal{O}^2_{\mathsf{G},b,\alpha,\boldsymbol{h}}(\cdot)}(\mathsf{st}),\end{aligned}$$

where each oracle $\mathcal{O}^1,\mathcal{O}^2$ *can be queried at most t_1,t_2 times respectively*, and is defined as follows.

- **Selective Master-key Hiding Security.**
  - $\mathcal{O}^1_{\mathsf{SMH},b,\alpha,\boldsymbol{h}}(Y)$: Run $(\boldsymbol{c};w_2)\leftarrow\mathsf{Enc2}(Y,p_2)$; $\boldsymbol{s}\xleftarrow{\$}\mathbb{Z}_{p_2}^{(w_2+1)}$; return $\boldsymbol{C}\leftarrow g_2^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}$.
  - $\mathcal{O}^2_{\mathsf{SMH},b,\alpha,\boldsymbol{h}}(X)$: If $R_{p_2}(X,Y)=1$ for some Y queried to $\mathcal{O}^1$, then return $\perp$. Else, run $(\boldsymbol{k};m_2)\leftarrow\mathsf{Enc1}(X,p_2)$; $\boldsymbol{r}\xleftarrow{\$}\mathbb{Z}_{p_2}^{m_2}$; return $\boldsymbol{K}\leftarrow\begin{cases}g_2^{\boldsymbol{k}(0,\boldsymbol{r},\boldsymbol{h})} & \text{if } b=0\\ g_2^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} & \text{if } b=1\end{cases}$.
- **Co-selective Master-key Hiding Security.**
  - $\mathcal{O}^1_{\mathsf{CMH},b,\alpha,\boldsymbol{h}}(X)$: Run $(\boldsymbol{k};m_2)\leftarrow\mathsf{Enc1}(X,p_2)$; $\boldsymbol{r}\xleftarrow{\$}\mathbb{Z}_{p_2}^{m_2}$; return $\boldsymbol{K}\leftarrow\begin{cases}g_2^{\boldsymbol{k}(0,\boldsymbol{r},\boldsymbol{h})} & \text{if } b=0\\ g_2^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} & \text{if } b=1\end{cases}$.
  - $\mathcal{O}^2_{\mathsf{CMH},b,\alpha,\boldsymbol{h}}(Y)$: If $R_{p_2}(X,Y)=1$ for some X queried to $\mathcal{O}^1$, then return $\perp$. Else, run $(\boldsymbol{c};w_2)\leftarrow\mathsf{Enc2}(Y,p_2)$; $\boldsymbol{s}\xleftarrow{\$}\mathbb{Z}_{p_2}^{(w_2+1)}$; return $\boldsymbol{C}\leftarrow g_2^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}$.

We define the advantage of $\mathcal{A}$ against the pair encoding scheme P in the security game $\mathsf{G} \in \{\mathsf{SMH}, \mathsf{CMH}\}$ for bilinear group generator $\mathcal{G}$ with the bounded number of queries (t_1, t_2) as

$$\mathsf{Adv}_{\mathcal{A}}^{(t_1,t_2)\text{-}\mathsf{G}(\mathsf{P})}(\lambda) := |\Pr[\mathsf{Exp}_{\mathcal{G},\mathsf{P},\mathsf{G},0,\mathcal{A},t_1,t_2}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathcal{G},\mathsf{P},\mathsf{G},1,\mathcal{A},t_1,t_2}(\lambda) = 1]|$$

We say that P is (t_1, t_2)-*selectively master-key hiding* in $\mathcal{G}$ if $\mathsf{Adv}_{\mathcal{A}}^{(t_1,t_2)\text{-}\mathsf{SMH}(\mathsf{P})}(\lambda)$ is negligible for all polynomial time attackers $\mathcal{A}$. Analogously, P is (t_1, t_2)-*co-selectively master-key hiding* in $\mathcal{G}$ if $\mathsf{Adv}_{\mathcal{A}}^{(t_1,t_2)\text{-}\mathsf{CMH}(\mathsf{P})}(\lambda)$ is negligible for all polynomial time attackers $\mathcal{A}$.

Poly-many Queries. We also consider the case where t_i is *not a-priori bounded* and hence the corresponding oracle can be queried polynomially many times. In such a case, we denote t_i as poly.

Remark 1 (Relation to Notions in [2])**.** The original notions considered in [2] are $(1, \mathsf{poly})$-SMH, $(1, 1)$-CMH for selective and co-selective master-key hiding security, respectively. In this paper, our conversion will convert a $(1, 1)$-SMH-secure pair encoding scheme into another scheme which is $(1, 1)$-CMH-secure, and vice-versa. We note that $(1, \mathsf{poly})$-SMH trivially implies $(1, 1)$-SMH.

We also refer the definition of *perfectly master-key hiding* to [2]. Informally, it requires α to be information-theoretically hidden from $\boldsymbol{c}_Y(\boldsymbol{s}, \boldsymbol{h})$, $\boldsymbol{k}_X(\alpha, \boldsymbol{r}, \boldsymbol{h})$ for any X, Y such that $R(X, Y) = 0$.

3.2 Implications to Fully Secure ABE

From a pair encoding scheme P for R, an ABE scheme for R, denoted $\mathsf{ABE}(\mathsf{P})$, can be achieved via the generic construction of [2], which we recall it here.

- $\mathsf{Setup}(1^\lambda, \kappa)$: Run $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Pick generators $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}$, $Z_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Obtain $n \leftarrow \mathsf{Param}(\kappa)$. Pick $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$ and $\alpha \xleftarrow{\$} \mathbb{Z}_N$. The public key is $\mathsf{PK} = \big(g_1, e(g_1, g_1)^\alpha, g_1^{\boldsymbol{h}}, Z_3\big)$. The master secret key is $\mathsf{MSK} = \alpha$.

- $\mathsf{Encrypt}(Y, M, \mathsf{PK})$: Upon input $Y \in \mathbb{Y}_N$, run $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, N)$. Then pick $\boldsymbol{s} = (s_0, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Output a ciphertext $\mathsf{CT} = (\boldsymbol{C}, C_0)$ where

$$\boldsymbol{C} = g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})} \in \mathbb{G}^{w_1}, C_0 = (e(g_1, g_1)^\alpha)^{s_0} M \in \mathbb{G}_T.$$

  Note that $\boldsymbol{C}$ can be computed from $g_1^{\boldsymbol{h}}$ and $\boldsymbol{s}$ since $\boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h})$ contains only linear combinations of monomials $s_i, sh_j, s_i h_j$.

- $\mathsf{KeyGen}(X, \mathsf{MSK}, \mathsf{PK})$: Upon input $X \in \mathbb{X}_N$, run $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, N)$. Parse $\mathsf{MSK} = \alpha$. Recall that $m_1 = |\boldsymbol{k}|$. Pick $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$. Output a secret key

$$\mathsf{SK} = g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot \boldsymbol{R}_3 \in \mathbb{G}^{m_1}.$$

- $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK})$: Parse Y, X from CT, SK. Assume $R(X, Y) = 1$. Run $\boldsymbol{E} \leftarrow \mathsf{Pair}(X, Y)$. Compute $e(g_1, g_1)^{\alpha s_0} \leftarrow e(\boldsymbol{K}^{\boldsymbol{E}}, \boldsymbol{C})$, and $M \leftarrow C_0 / e(g_1, g_1)^{\alpha s_0}$.

Its correctness follows from that of the pair encoding, see [2]. Also in [2], it is proved that if P is $(1, \mathsf{poly})$-SMH and $(1,1)$-CMH secure, then $\mathsf{ABE}(\mathsf{P})$ is fully secure with reduction $O(q_1)$. We recall this as follows. Let $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}(\mathsf{P})}(\lambda)$ be the advantage of an adversary $\mathcal{A}$ against the full security of $\mathsf{ABE}(\mathsf{P})$.

Proposition 1 ([2])**.** *Suppose that a pair encoding* P *for predicate* R *is both* $(1,1)$-CMH *and* $(1, \mathsf{poly})$-SMH *in* $\mathcal{G}$*. Suppose that the Subgroup Decision Assumption 1,2,3 (denoted as* $\mathsf{SD1}, \mathsf{SD2}, \mathsf{SD3}$*)*[2] *hold in* $\mathcal{G}$*. Suppose also that* R *is domain-transferable.*[3] *Then the ABE scheme* $\mathsf{ABE}(\mathsf{P})$ *in* $\mathcal{G}$ *for predicate* R *is fully secure. More precisely, for any PPT adversary* $\mathcal{A}$*, there exist PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5$*, whose running times are the same as* $\mathcal{A}$ *plus some polynomial times, such that for any* λ*,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}(\mathsf{P})}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SD1}}(\lambda) + (2q_1+3)\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{SD2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SD3}}(\lambda) \\ + q_1\mathsf{Adv}_{\mathcal{B}_4}^{(1,1)\text{-}\mathsf{CMH}(\mathsf{P})}(\lambda) + \mathsf{Adv}_{\mathcal{B}_5}^{(1,\mathsf{poly})\text{-}\mathsf{SMH}(\mathsf{P})}(\lambda),$$

where q_1 *is the number of queries in phase 1.*

As a new corollary, we have that if P is $(1,1)$-SMH and $(1,1)$-CMH secure, then $\mathsf{ABE}(\mathsf{P})$ is fully secure with reduction $O(q_{\mathrm{all}})$. We state this as follows.

Corollary 2. *Suppose that a pair encoding scheme* P *for predicate* R *is both* $(1,1)$-CMH *and* $(1,1)$-SMH *in* $\mathcal{G}$*. Suppose that* $\mathsf{SD1}, \mathsf{SD2}, \mathsf{SD3}$ *hold in* $\mathcal{G}$*. Suppose also that* R *is domain-transferable. Then,* $\mathsf{ABE}(\mathsf{P})$ *in* $\mathcal{G}$ *for predicate* R *is fully secure. More precisely, for any PPT adversary* $\mathcal{A}$*, there exist PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5$*, whose running times are the same as* $\mathcal{A}$ *plus some poly times, such that for any* λ*,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}(\mathsf{P})}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SD1}}(\lambda) + (2q_{\mathrm{all}}+1)\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{SD2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SD3}}(\lambda) \\ + q_1\mathsf{Adv}_{\mathcal{B}_4}^{(1,1)\text{-}\mathsf{CMH}(\mathsf{P})}(\lambda) + q_2\mathsf{Adv}_{\mathcal{B}_5}^{(1,1)\text{-}\mathsf{SMH}(\mathsf{P})}(\lambda),$$

where q_1 *and* q_2 *denotes the number of queries in phase 1 and phase 2, respectively, and* $q_{\mathrm{all}} = q_1 + q_2$*.*

Proof of Corollary 2 (Sketch). This corollary follows the proof of Proposition 1 in [2]. The only difference is that instead of switching all *post-challenge* keys *all at once* for the three games (normal to semi-functional type 1, to type 2, and to type 3), we switch each post-challenge key *one key per one game*, in just the same way as for each pre-challenge key (and as in the traditional dual system encryption proofs). This results in the cost q_2 for the reduction to the SMH security and the additional cost $2q_2 - 2$ for the reduction to $\mathsf{SD2}$. □

4 New Theorem for Generic Dual Conversion

In this section, we first recall the generic dual conversion of [2], where it was proved to hold for only the case of perfectly master-key hiding encoding. We restate this as Proposition 3.

[2] The Subgroup Decision Assumptions $\mathsf{SD1}, \mathsf{SD2}, \mathsf{SD3}$ were introduced in [25] as main ingredients of dual system encryption approaches based on composite-order groups. We refer their definitions to [25, 2].

[3] Informally speaking, R is domain-transferable [2] if $R_N(X,Y) = R_p(X,Y)$ for any prime $p|N$ with high probability.

We then present our main results which are new theorems for the case of computationally secure encodings.

Dual Conversion of [2]. Given a pair encoding scheme P_R for predicate R, we construct a predicate encoding scheme $\mathcal{C}(\mathsf{P}_R)$ for $\bar{R}$ as follows. For $\mathsf{Param} \to (n, \boldsymbol{h})$, we set $\overline{\mathsf{Param}} = (n+1, \overline{\boldsymbol{h}})$ where $\overline{\boldsymbol{h}} = (\boldsymbol{h}, \bar{\phi})$, where $\bar{\phi}$ is a new variable. We then define

- $\overline{\mathsf{Enc1}}(X, N)$: Obtain $(\boldsymbol{c}_X(\boldsymbol{s}, \boldsymbol{h}); w_2) \leftarrow \mathsf{Enc2}(X, N)$ and parse $\boldsymbol{s} = (s_0, \ldots)$. Then, set
$$\bar{\boldsymbol{k}}_X(\bar{\alpha}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{h}}) := \big(\boldsymbol{c}_X(\boldsymbol{s}, \boldsymbol{h}),\ \bar{\alpha} + \bar{\phi} s_0\big), \qquad\qquad \bar{\boldsymbol{r}} := \boldsymbol{s},$$
and output $(\bar{\boldsymbol{k}}_X(\bar{\alpha}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{h}}); w_2)$, where we treat $\bar{\alpha}$ as a new variable.
- $\overline{\mathsf{Enc2}}(Y, N)$: Obtain $(\boldsymbol{k}_Y(\alpha, \boldsymbol{r}, \boldsymbol{h}); m_2) \leftarrow \mathsf{Enc1}(Y, N)$. Then, set
$$\bar{\boldsymbol{c}}_Y(\bar{\boldsymbol{s}}, \bar{\boldsymbol{h}}) := \big(\boldsymbol{k}_Y(\bar{\phi}\bar{s}_0, \boldsymbol{r}, \boldsymbol{h}),\ \bar{s}_0\big), \qquad\qquad \bar{\boldsymbol{s}} := (\bar{s}_0, \boldsymbol{r}),$$
and output $(\bar{\boldsymbol{c}}_Y(\bar{\boldsymbol{s}}, \bar{\boldsymbol{h}}); m_2)$, where we treat $\bar{s}_0$ as a new variable.

The correctness can be verified as follows. If $\bar{R}(X, Y) = 1$, then $R(Y, X) = 1$, hence from $\boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h})$ and $\boldsymbol{k}(\bar{\phi}\bar{s}_0, \boldsymbol{r}, \boldsymbol{h})$, we can compute $(\bar{\phi}\bar{s}_0)s_0$, thanks to the correctness of P_R. From that, we obtain $(\alpha + \bar{\phi}s_0)(\bar{s}_0) - (\bar{\phi}\bar{s}_0)s_0 = \alpha\bar{s}_0$. We also note that $\mathcal{C}(\mathsf{P}_R)$ is normal by definition.

Proposition 3 ([2])**.** *If the pair encoding P_R for R is* perfectly *master-key hiding, then the pair encoding $\mathcal{C}(\mathsf{P}_R)$ for $\bar{R}$ is also* perfectly *master-key hiding.*

Theorem 4. *If the pair encoding P_R for R is normal and* $(1,1)$-co-selectively *master-key hiding, then the pair encoding $\mathcal{C}(\mathsf{P}_R)$ for $\bar{R}$ is* $(1,1)$-selectively *master-key hiding (with tight reduction).*

Theorem 5. *If the pair encoding P_R for R is normal and* $(1,1)$-selectively *master-key hiding, then the pair encoding $\mathcal{C}(\mathsf{P}_R)$ for $\bar{R}$ is* $(1,1)$-co-selectively *master-key hiding (with tight reduction).*

Proof of Theorem 4. Suppose that there is an adversary $\mathcal{A}$ against the $(1,1)$-SMH security of $\mathcal{C}(\mathsf{P}_R)$. We construct an algorithm $\mathcal{B}$ against the $(1,1)$-CMH security of P_R as follows. At the initialization, $\mathcal{B}$ first obtains g_1, g_2, g_3 from its challenger. $\mathcal{B}$ simply parses these to $\mathcal{A}$ for initialization.

(Simulating $\mathcal{O}^1$). In the $(1,1)$-SMH game, $\mathcal{A}$ first makes a ciphertext query for Y. $\mathcal{B}$ then makes a key query for Y to its challenger in its own $(1,1)$-CMH game and obtains $\boldsymbol{K} = g_2^{\boldsymbol{k}_Y(\alpha, \boldsymbol{r}, \boldsymbol{h})}$. The goal of $\mathcal{B}$ is to guess if $\alpha = 0$ or $\alpha \in_R \mathbb{Z}_N$. $\mathcal{B}$ samples $\bar{\phi}', \bar{s}_0 \xleftarrow{\$} \mathbb{Z}_N$ and *implicitly* defines $\bar{\phi} = \bar{\phi}' + \alpha/\bar{s}_0$. $\mathcal{B}$ then computes

$$\tilde{\boldsymbol{C}} = g_2^{\boldsymbol{k}_Y(\bar{\phi}'\bar{s}_0, \mathbf{0}, \mathbf{0})} \cdot \boldsymbol{K} = g_2^{\boldsymbol{k}_Y(\bar{\phi}'\bar{s}_0 + \alpha, \boldsymbol{r}, \boldsymbol{h})} = g_2^{\boldsymbol{k}_Y(\bar{\phi}\bar{s}_0, \boldsymbol{r}, \boldsymbol{h})},$$

where the middle equation holds from the definition of $\boldsymbol{K}$ and thanks to the identity Eq. (1), while the last equation holds due to that $\bar{\phi}\bar{s}_0 = (\bar{\phi}' + \alpha/\bar{s}_0)\bar{s}_0 = \bar{\phi}'\bar{s}_0 + \alpha$. $\mathcal{B}$ then returns the ciphertext

$$\bar{\boldsymbol{C}} = \big(\tilde{\boldsymbol{C}}, g_2^{\bar{s}_0}\big) = g_2^{\bar{\boldsymbol{c}}_Y(\bar{\boldsymbol{s}}, \bar{\boldsymbol{h}})}$$

to $\mathcal{A}$. This perfectly simulates the answer for the query Y to $\mathcal{O}^1$ for $\mathcal{A}$.

(Simulating $\mathcal{O}^2$). $\mathcal{A}$ makes a key query for X such that $\bar{R}(X,Y) = 0$. $\mathcal{B}$ then makes a ciphertext query for X to its challenger in its own $(1,1)$-CMH game, which can be done since $R(Y,X) = \bar{R}(X,Y) = 0$, and obtains $\boldsymbol{C} = g_2^{\boldsymbol{c}_X(\boldsymbol{s},\boldsymbol{h})}$. $\mathcal{B}$ then *implicitly* defines $\bar{\alpha} = -\alpha s_0/\bar{s}_0$. This is distributed independently from other elements since the other place where α appears is in $\bar{\phi}$ but there, α is hidden by the random value $\bar{\phi}'$. $\mathcal{B}$ then computes

$$g_2^{\bar{\alpha}+\bar{\phi}s_0} = g_2^{(-\alpha s_0/\bar{s}_0)+(\bar{\phi}'+\alpha/\bar{s}_0)s_0} = g_2^{-\alpha s_0/\bar{s}_0+\bar{\phi}' s_0+\alpha s_0/\bar{s}_0} = g_2^{\bar{\phi}' s_0},$$

which can be computed since $g_2^{s_0}$ is available from $\boldsymbol{C}$ due to the *normality* of encoding. $\mathcal{B}$ returns

$$\bar{\boldsymbol{K}} = \left(\boldsymbol{C}, g_2^{\bar{\alpha}+\bar{\phi}s_0}\right) = g_2^{\bar{\boldsymbol{k}}_X(\bar{\alpha},\bar{\boldsymbol{r}},\bar{\boldsymbol{h}})}$$

to $\mathcal{A}$. It perfectly simulates the answer for the query X to $\mathcal{O}^2$ for $\mathcal{A}$.

(Output). Finally, when $\mathcal{A}$ outputs b' as its guess, $\mathcal{B}$ also outputs the same value b'. Now since we have (implicitly) defined $\bar{\alpha} = -\alpha s_0/\bar{s}_0$, we have that if $\alpha = 0$, then $\bar{\alpha} = 0$, and if $\alpha \in_R \mathbb{Z}_N$, then $\bar{\alpha} \in_R \mathbb{Z}_N$. Therefore, the advantage of $\mathcal{B}$ is equal to that of $\mathcal{A}$. This concludes the proof. □

Proof of Theorem 5. Suppose that there is an adversary $\mathcal{A}$ against the $(1,1)$-CMH security of $\mathcal{C}(\mathsf{P}_R)$. We claim that we can construct an efficient algorithm $\mathcal{B}$ against the $(1,1)$-SMH security of P_R that has the same advantage as $\mathcal{A}$, and hence conclude the proof. This can be done analogously to the previous proof. The only difference is the order of the key and ciphertext queries by A. In the $(1,1)$-CMH game, $\mathcal{A}$ makes a key query for Y first, then a ciphertext query for X. But this is exactly the same order in the $(1,1)$-SMH game for $\mathcal{B}$, where $\mathcal{B}$ will ask a ciphertext query for Y first, then a key query for X. The detailed simulation is exactly the same as the previous proof. □

The following corollary follows from the above two theorems and Corollary 2.

Corollary 6. *For any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5$, whose running times are the same as $\mathcal{A}$ plus some polynomial times, such that for any λ,*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}(\mathcal{C}(\mathsf{P}))}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SD1}}(\lambda) + (2q_{\mathrm{all}}+1)\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{SD2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SD3}}(\lambda) \\ + q_1\mathsf{Adv}_{\mathcal{B}_4}^{(1,1)\text{-}\mathsf{SMH(P)}}(\lambda) + q_2\mathsf{Adv}_{\mathcal{B}_5}^{(1,1)\text{-}\mathsf{CMH(P)}}(\lambda),\end{aligned}$$

where q_1 and q_2 denotes the number of queries in phase 1 and phase 2, respectively, and $q_{\mathrm{all}} = q_1 + q_2$.

5 Concrete Dual Schemes with Tighter Reduction

Our generic dual conversion in the previous section can convert $(1,1)$-CMH-secure encoding into $(1,1)$-SMH-secure encoding, and vice versa. This results in ABE with $O(q_{\mathrm{all}})$ reduction by Corollary 2. In this section, we provide a direct construction of pair encoding scheme of a certain dual predicate and show that it is $(1,1)$-CMH-secure and $(1,\mathsf{poly})$-SMH-secure.

Therefore, the resulting ABE enjoys tighter reduction of $O(q_1)$ by Proposition 1. We focus on the CP-DSE primitive, which is the dual of KP-DSE. Although we will obtain a specific scheme, we give a generic conversion that is extended from the previous conversion. This conversion has the same properties as in Theorem 4 and 5, that is, it converts $(1,1)$-CMH-secure encoding into $(1,1)$-SMH-secure encoding, and vice versa. The new result here is that we can prove the $(1,\mathsf{poly})$-SMH security of the encoding scheme for CP-DSE obtained by applying this new conversion to the encoding of KP-DSE in [2]. Intuitively, we use the *randomizer technique* from [27, 2] for obtaining $(1,\mathsf{poly})$-SMH. To enable this, we require one more element each for a key and a ciphertext (elements related to $\bar{u},\bar{\eta}$ below).

Extended Dual Conversion. Given a pair encoding scheme P_R for predicate R, we construct a predicate encoding scheme $\mathcal{EC}(\mathsf{P}_R)$ for $\bar{R}$ as follows. For $\mathsf{Param} \to (n,\boldsymbol{h})$, we set $\overline{\mathsf{Param}} = (n+2,\bar{\boldsymbol{h}})$ where $\bar{\boldsymbol{h}} = (\boldsymbol{h},\bar{\phi},\bar{\eta})$, where $\bar{\phi},\bar{\eta}$ are new variables. We then define

- $\overline{\mathsf{Enc1}}(X,N)$: Obtain $(\boldsymbol{c}_X(\boldsymbol{s},\boldsymbol{h});w_2) \leftarrow \mathsf{Enc2}(X,N)$ and parse $\boldsymbol{s}=(s_0,\ldots)$. Then, set
$$\bar{\boldsymbol{k}}_X(\bar{\alpha},\bar{\boldsymbol{r}},\bar{\boldsymbol{h}}) := \big(\boldsymbol{c}_X(\boldsymbol{s},\boldsymbol{h}),\ \bar{\alpha}+\bar{\phi}s_0+\bar{u}\bar{\eta},\ \bar{u}\big), \qquad \bar{\boldsymbol{r}} := (\boldsymbol{s},\bar{u}),$$
and output $(\bar{\boldsymbol{k}}_X(\bar{\alpha},\bar{\boldsymbol{r}},\bar{\boldsymbol{h}});w_2+1)$, where we treat $\bar{\alpha},\bar{u}$ as new variables.

- $\overline{\mathsf{Enc2}}(Y,N)$: Obtain $(\boldsymbol{k}_Y(\alpha,\boldsymbol{r},\boldsymbol{h});m_2) \leftarrow \mathsf{Enc1}(Y,N)$. Then, set
$$\bar{\boldsymbol{c}}_Y(\bar{\boldsymbol{s}},\bar{\boldsymbol{h}}) := \big(\boldsymbol{k}_Y(\bar{\phi}\bar{s}_0,\boldsymbol{r},\boldsymbol{h}),\ \bar{s}_0,\ \bar{s}_0\bar{\eta}\big), \qquad \bar{\boldsymbol{s}} := (\bar{s}_0,\boldsymbol{r}),$$
and output $(\bar{\boldsymbol{c}}_Y(\bar{\boldsymbol{s}},\bar{\boldsymbol{h}});m_2)$, where we treat $\bar{s}_0$ as a new variable.

The correctness can be verified as follows. If $\bar{R}(X,Y)=1$, then $R(Y,X)=1$, hence from $\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})$ and $\boldsymbol{k}(\bar{\phi}\bar{s}_0,\boldsymbol{r},\boldsymbol{h})$, we can compute $(\bar{\phi}\bar{s}_0)s_0$, thanks to the correctness of P_R. We thus obtain $(\alpha+\bar{\phi}s_0+\bar{u}\bar{\eta})(\bar{s}_0)-(\bar{\phi}\bar{s}_0)s_0-\bar{u}(\bar{s}_0\bar{\eta})=\alpha\bar{s}_0$.

Corollary 7. *If the pair encoding scheme P_R for R is $(1,1)$-CMH, then the pair encoding scheme $\mathcal{EC}(\mathsf{P}_R)$ for $\bar{R}$ is $(1,1)$-SMH. If the pair encoding scheme P_R for R is $(1,1)$-SMH, then the pair encoding scheme $\mathcal{EC}(\mathsf{P}_R)$ for $\bar{R}$ is $(1,1)$-CMH.*

Proof. The proof follows exactly in the same manner as Theorem 4, 5 except that the reduction $\mathcal{B}$ also randomly chooses $\bar{\eta},\bar{u}$. The corresponding terms can be computed using $\bar{\eta},\bar{u}$. □

5.1 Our CP-DSE with Tighter Reduction

For ease of reading, we defer the definition of CP-DSE and our concrete construction for it, which involve in more details, to §A. Roughly speaking, we obtain the following result. Let $\mathsf{P}_{\mathsf{KPDSE}}$ denote the pair encoding construction for KP-DSE of [2]. We obtain a new pair encoding for CP-DSE as $\mathcal{EC}(\mathsf{P}_{\mathsf{KPDSE}})$. We prove that it is $(1,\mathsf{poly})$-SMH-secure with tight reduction under a new assumption which is similar to the assumption use for proving the CMH security of $\mathsf{P}_{\mathsf{KPDSE}}$ of [2].

6 Generic Conjunction and Conversion to Dual Policy

Let $R_1 : \mathbb{X}_1 \times \mathbb{Y}_1$, $R_2 : \mathbb{X}_2 \times \mathbb{Y}_2$ be two predicates. We define the *conjunctive predicate* of R_1, R_2 as $[R_1 \wedge R_2] : \tilde{\mathbb{X}} \times \tilde{\mathbb{Y}} \to \{0,1\}$ where $\tilde{\mathbb{X}} = \mathbb{X}_1 \times \mathbb{X}_2$, $\tilde{\mathbb{Y}} = \mathbb{Y}_1 \times \mathbb{Y}_2$ and $[R_1 \wedge R_2]((X_1, X_2), (Y_1, Y_2)) = 1$ iff $R_1(X_1, Y_1) = 1$ *and* $R_2(X_2, Y_2) = 1$.

Next, let $R : \mathbb{X} \times \mathbb{Y}$ be a predicate. We define its *dual-policy predicate* (DP) as the conjunctive of itself and its dual predicate, $\bar{R}$. Hence, its notation is $[R \wedge \bar{R}]$.

Conjunctive Predicate Conversion. Given two pair encoding schemes: P_{R_1} for predicate R_1 and P_{R_2} for predicate R_2, we construct a predicate encoding scheme denoted $\mathcal{D}(\mathsf{P}_{R_1}, \mathsf{P}_{R_2})$ for predicate $[R_1 \wedge R_2]$ as follows. For $\mathsf{Param}_1 \to (n_1, \boldsymbol{h}_1)$, $\mathsf{Param}_2 \to (n_2, \boldsymbol{h}_2)$, we set $\widehat{\mathsf{Param}} = (n_1 + n_2, \hat{\boldsymbol{h}})$ where $\hat{\boldsymbol{h}} = (\boldsymbol{h}_1, \boldsymbol{h}_2)$. We then define

- $\widehat{\mathsf{Enc1}}((X_1, X_2), N)$: For $i = 1, 2$, obtain $(\boldsymbol{k}_{X_i}(\alpha_i, \boldsymbol{r}_i, \boldsymbol{h}_i); m_{2,i}) \leftarrow \mathsf{Enc1}_i(X_i, N)$. Then, set

  $$\hat{\boldsymbol{k}}_{(X_1,X_2)}(\hat{\alpha}, \hat{\boldsymbol{r}}, \hat{\boldsymbol{h}}) := \big(\boldsymbol{k}_{X_1}(\hat{r}, \boldsymbol{r}_1, \boldsymbol{h}_1), \boldsymbol{k}_{X_2}(\hat{\alpha} - \hat{r}, \boldsymbol{r}_2, \boldsymbol{h}_2)\big), \qquad \hat{\boldsymbol{r}} := (\boldsymbol{r}_1, \boldsymbol{r}_2, \hat{r}),$$

  and output $(\hat{\boldsymbol{k}}_X(\alpha, \hat{\boldsymbol{r}}, \hat{\boldsymbol{h}}); m_{2,1} + m_{2,2} + 1)$, where we treat $\hat{\alpha}, \hat{r}$ as new variables.

- $\widehat{\mathsf{Enc2}}(Y, N)$: For $i = 1, 2$, obtain $(\boldsymbol{c}_{Y_i}(\boldsymbol{s}_i, \boldsymbol{h}_i); w_{2,i}) \leftarrow \mathsf{Enc2}_i(Y_i, N)$. Parse $\boldsymbol{s}_i = (s_{0,i}, \boldsymbol{s}'_i)$, and set

  $$\hat{\boldsymbol{c}}_{(Y_1,Y_2)}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{h}}) := \big(\boldsymbol{c}_{Y_1}((s_0, \boldsymbol{s}'_1), \boldsymbol{h}_1), \boldsymbol{c}_{Y_2}((s_0, \boldsymbol{s}'_2), \boldsymbol{h}_2)\big), \qquad \hat{\boldsymbol{s}} := (s_0, \boldsymbol{s}'_1, \boldsymbol{s}'_2),$$

  and output $(\hat{\boldsymbol{c}}_{(Y_1,Y_2)}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{h}}); w_{2,1} + w_{2,2})$, where we treat s_0 as a new variable.

The correctness can be verified as follows. If $[R_1 \wedge R_2]((X_1, X_2), (Y_1, Y_2)) = 1$, then we have $R_1(X_1, Y_1) = 1$ *and* $R_2(X_2, Y_2) = 1$. Hence, from $\boldsymbol{k}_{X_1}(\hat{r}, \boldsymbol{r}_1, \boldsymbol{h}_1)$ and $\boldsymbol{c}_{Y_1}((s_0, \boldsymbol{s}'_1), \boldsymbol{h}_1)$, we obtain $\hat{r}s_0$, due to the correctness of P_{R_1}. Similarly, from $\boldsymbol{k}_{X_2}(\hat{\alpha} - \hat{r}, \boldsymbol{r}_2, \boldsymbol{h}_2)$ and $\boldsymbol{c}_{Y_2}((s_0, \boldsymbol{s}'_2), \boldsymbol{h}_2)$, we obtain $(\hat{\alpha} - \hat{r})s_0$, due to the correctness of P_{R_2}. From these, we obtain $\hat{r}s_0 + (\hat{\alpha} - \hat{r})s_0 = \hat{\alpha}s_0$.

Theorem 8. *If the pair encoding schemes P_{R_1} for R_1 and P_{R_2} for R_2 are perfectly master-key hiding, then the pair encoding scheme $\mathcal{D}(\mathsf{P}_{R_1}, \mathsf{P}_{R_2})$ for $[R_1 \wedge R_2]$ is also perfectly master-key hiding.*

Proof. Consider $(X_1, X_2), (Y_1, Y_2)$ such that $[R_1 \wedge R_2]((X_1, X_2), (Y_1, Y_2)) = 0$. If $R_1(X_1, Y_1) = 0$, from the perfect security of P_{R_1}, we have that $\hat{r}$ is hidden, hence $\hat{\alpha}$ is also hidden since it is masked with $\hat{r}$. If $R_2(X_2, Y_2) = 0$, from the perfect security of P_{R_2}, we have $\hat{\alpha} - \hat{r}$ is hidden and hence $\hat{\alpha}$ is also hidden. In both cases, we have that $\hat{\alpha}$ is hidden as required. □

Theorem 9. *For the notion $\mathsf{X} \in \{(1,1)\text{-}\mathsf{SMH}, (1,1)\text{-}\mathsf{CMH}\}$, if the pair encoding schemes P_{R_1} for R_1 and P_{R_2} for R_2 are both normal and X-secure, then the pair encoding scheme $\mathcal{D}(\mathsf{P}_{R_1}, \mathsf{P}_{R_2})$ for $[R_1 \wedge R_2]$ is also X-secure. More precisely, for any PPT adversary $\mathcal{A}$, there exist a PPT algorithm $\mathcal{B}$, whose running time is the same as $\mathcal{A}$ plus some polynomial time, such that for any λ,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{X}(\mathcal{D}(\mathsf{P}_{R_1}, \mathsf{P}_{R_2}))}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{B}}^{\mathsf{X}(\mathsf{P}_{R_1})}(\lambda) + 2\mathsf{Adv}_{\mathcal{B}}^{\mathsf{X}(\mathsf{P}_{R_2})}(\lambda) \tag{2}$$

The following corollary is immediate from Theorem 4, 5, and 9.

Corollary 10. *If the pair encoding scheme* P_R *for* R *is normal,* $(1,1)$-selectively*, and* $(1,1)$-co-selectively *master-key hiding, then the pair encoding* $\mathcal{D}(\mathsf{P}_R, \mathcal{C}(\mathsf{P}_R))$ *for* $[R \wedge \bar{R}]$ *is also* $(1,1)$-selectively *and* $(1,1)$-co-selectively *master-key hiding.*

Proof of Theorem 9. We prove for the case of SMH. The case for CMH can be done in exactly the same manner except exchanging the order of oracles. Suppose that there is an adversary $\mathcal{A}$ against the $(1,1)$-SMH security of $\mathcal{D}(\mathsf{P}_{R_1}, \mathsf{P}_{R_2})$. We construct an algorithm $\mathcal{B}$ against the $(1,1)$-SMH security of either P_{R_1} or P_{R_2} as follows. Firstly, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{1,2\}$ for determining to break the $(1,1)$-SMH security of P_{R_b}. At the initialization, $\mathcal{B}$ first obtains g_1, g_2, g_3 from its challenger (of the $(1,1)$-SMH game for P_{R_b}). $\mathcal{B}$ simply parses these to $\mathcal{A}$ for initialization. Let $\tilde{b} = 1$ if $b = 2$, and $\tilde{b} = 2$ if $b = 1$. $\mathcal{B}$ will construct all parameters for $\mathsf{P}_{R_{\tilde{b}}}$ by itself by choosing $\boldsymbol{h}_{\tilde{b}} \xleftarrow{\$} \mathbb{Z}_p^{n_{\tilde{b}}}$.

(Simulating $\mathcal{O}^1$). In the $(1,1)$-SMH game, $\mathcal{A}$ first makes a ciphertext query for (Y_1, Y_2). $\mathcal{B}$ then makes a key query for Y_b to its challenger (of the $(1,1)$-SMH game for P_{R_b}) and obtains $g_2^{\boldsymbol{c}_{Y_b}(\boldsymbol{s}_b, \boldsymbol{h}_b)}$. Due to the normality, $\mathcal{B}$ can parse $g_2^{s_{0,b}}$ from this. We implicitly set $s_0 = s_{0,b}$. $\mathcal{B}$ chooses $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_p^{w_{2,\tilde{b}}}$ then computes

$$(g_2^{s_0})^{\boldsymbol{c}_{Y_{\tilde{b}}}((1,\boldsymbol{\delta}),\boldsymbol{h}_{\tilde{b}})} = g_2^{\boldsymbol{c}_{Y_{\tilde{b}}}\left((s_0,\, s_0\boldsymbol{\delta}),\boldsymbol{h}_{\tilde{b}}\right)},$$

which holds due to linearity. This implicitly sets $\boldsymbol{s}'_{\tilde{b}} = s_0\boldsymbol{\delta}$. The algorithm $\mathcal{B}$ then returns the pair $g_2^{\boldsymbol{c}_{Y_b}(\boldsymbol{s}_b, \boldsymbol{h}_b)}$ and $g_2^{\boldsymbol{c}_{Y_{\tilde{b}}}((s_0, \boldsymbol{s}'_{\tilde{b}}), \boldsymbol{h}_{\tilde{b}})}$ in the order according to b (*i.e.*, if $b = 1$, they are in this order, otherwise, we swap them).

(Simulating $\mathcal{O}^2$). The adversary $\mathcal{A}$ makes a key query for (X_1, X_2) such that $[R_1 \wedge R_2]((X_1, X_2), (Y_1, Y_2)) = 0$. There are two possible cases. If $R_b(X_b, Y_b) = 0$, then $\mathcal{B}$ makes a key query for X_b to its challenger (of the $(1,1)$-SMH game for P_{R_b}) and obtains $\boldsymbol{K}_b = g_2^{\boldsymbol{k}_{X_b}(\alpha_b, \boldsymbol{r}_b, \boldsymbol{h}_b)}$. Otherwise, $R_b(X_b, Y_b) = 1$, $\mathcal{B}$ will ask some legitimate key query and simply outputs a random guess, while abort the game with $\mathcal{A}$. We now proceed with the former case, where it is further categorized into two cases:

- If $b = 1$, then $\mathcal{B}$ implicitly sets $\hat{\alpha} = \alpha_1$ and $\hat{r} = \alpha_1 + \hat{r}'$ where $\mathcal{B}$ chooses $\hat{r}' \xleftarrow{\$} \mathbb{Z}_p$. Hence, $\hat{\alpha} - \hat{r} = -\hat{r}'$. $\mathcal{B}$ computes

  $$\hat{\boldsymbol{K}}_1 := g_2^{\boldsymbol{k}_{X_1}(\hat{r}', \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{K}_1 = g_2^{\boldsymbol{k}_{X_1}(\alpha_1 + \hat{r}', \boldsymbol{r}_1, \boldsymbol{h}_1)} = g_2^{\boldsymbol{k}_{X_1}(\hat{r}, \boldsymbol{r}_1, \boldsymbol{h}_1)},$$

  which holds from the identity Eq. (1). $\mathcal{B}$ also computes

  $$\hat{\boldsymbol{K}}_2 := g_2^{\boldsymbol{k}_{X_2}(-\hat{r}', \boldsymbol{r}_2, \boldsymbol{h}_2)} = g_2^{\boldsymbol{k}_{X_2}(\hat{\alpha} - \hat{r}, \boldsymbol{r}_2, \boldsymbol{h}_2)}$$

  by choosing $\boldsymbol{r}_2 \xleftarrow{\$} \mathbb{Z}_p^{m_{2,2}}$ (and recall that $\mathcal{B}$ possesses $\boldsymbol{h}_2$). $\mathcal{B}$ returns $(\hat{\boldsymbol{K}}_1, \hat{\boldsymbol{K}}_2)$ to $\mathcal{A}$.

- If $b = 2$, then $\mathcal{B}$ implicitly sets $\hat{\alpha} = \alpha_2$. $\mathcal{B}$ chooses $\hat{r} \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{B}$ computes

  $$\hat{\boldsymbol{K}}_1 := g_2^{\boldsymbol{k}_{X_1}(\hat{r}, \boldsymbol{r}_1, \boldsymbol{h}_1)}$$

  by choosing $\boldsymbol{r}_1 \xleftarrow{\$} \mathbb{Z}_p^{m_{2,1}}$ (and recall that $\mathcal{B}$ possesses $\boldsymbol{h}_1$). $\mathcal{B}$ then computes

  $$\hat{\boldsymbol{K}}_2 := g_2^{\boldsymbol{k}_{X_2}(-\hat{r}, \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{K}_2 = g^{\boldsymbol{k}_{X_2}(\hat{\alpha} - \hat{r}, \boldsymbol{r}_2, \boldsymbol{h}_2)},$$

  which holds due to the identity Eq. (1). $\mathcal{B}$ returns $(\hat{\boldsymbol{K}}_1, \hat{\boldsymbol{K}}_2)$ to $\mathcal{A}$.

In both cases, we have $\hat{\alpha} = \alpha_b$. Hence $\mathcal{B}$ just outputs its guess (of whether $\alpha_b = 0$ or $\alpha_b \xleftarrow{\$} \mathbb{Z}_p$) to be exactly the same as the output of $\mathcal{A}$ (who guesses whether $\hat{\alpha} = 0$ or $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_p$). Since $\mathcal{B}$ aborts with probability $1/2$, we have the inequality (2). □

7 Implied Instantiations

Policy over Doubly-Spatial Encryption. We obtain the first two (fully-secure) CP-DSE schemes. The first scheme is automatically obtained by applying the generic dual conversion to the KP-DSE of [2] (and use Theorem 4, 5). The resulting CP-DSE has reduction $O(q_{\text{all}})$, as shown in Corollary 6. The second scheme is directly constructed and has tighter reduction of $O(q_1)$ (see §5). We then obtain the first dual-policy over DSE (DP-DSE) by applying the generic conjunctive conversion to the KP-DSE of [2] and our first CP-DSE (and use Corollary 10).

ABE for Boolean Formulae (and Monotone Span Programs). We obtain various schemes:

- **Unbounded ABE.** We obtain the first fully-secure completely-unbounded CP-ABE schemes. Such schemes should pose no bounds such as the attribute set or policy size per ciphertext or key, the attribute universe size, and the number of attribute repetition (also called multi-use) in a policy. We use the fact that any pair encoding for CP-DSE implies an encoding for completely-unbounded CP-ABE as a special case. This is shown for the key-policy case in [2], but is also straightforward for the ciphertext-policy case by just exchanging key and ciphertext encodings. Hence, we have two completely-unbounded CP-ABE schemes, one with $O(q_{\text{all}})$ and one with $O(q_1)$ reduction. We then obtain the first completely-unbounded DP-ABE by applying the generic conjunctive conversion to the unbounded KP-ABE of [2] and our first unbounded CP-ABE (and use Corollary 10), or equivalently, we can view unbounded DP-ABE as a special case of DP-DSE.

- **ABE with Short Keys.** Any pair encoding for CP-DSE implies an encoding for CP-ABE with constant-size keys as a special case. This is analogous to the implication of KP-ABE with short ciphertexts from KP-DSE shown in [2]. We use the same implication but swap key and ciphertext encodings, hence short ciphertexts become short keys. From this, we obtain the first fully-secure CP-ABE with short keys. Note that it requires bounded-size attribute set per key.

- **(Bounded) ABE.** By applying the generic conjunctive conversion to the bounded KP-ABE and CP-ABE of [28] (and use Theorem 8), we obtain a fully-secure bounded DP-ABE for small-universe. Similarly, we obtain a large-universe variant from other KP-ABE and CP-ABE in [2] (namely, Scheme 12,13 in [2]). These systems require the bounds on the size of attribute sets for each ciphertext (in KP-ABE) or each key (in CP-ABE). Nevertheless, the underlying security of these encodings are perfectly master-key hiding, which is for free (no assumption needed for it), hence these systems use only subgroup decision assumptions required for the framework of [2].

ABE for Regular Languages (ABE-RL). In KP-ABE for regular languages, we have a key associated to the description of a deterministic finite automata (DFA) M, while a ciphertext is associated to a string w, and $R(M, w) = 1$ if the automata M accepts the string

Table 1: Previous schemes and our new instantiations, positioned by predicates and properties, where we recall that KP, CP, DP stands for key-policy, ciphertext-policy, and dual-policy, respectively.

Predicate	Properties			KP	CP	DP
	Security	Universe	Multi-use			
Policy over DSE	full	-	-	A14 [2]	**Ours**	**Ours**
Unbounded ABE	selective	large	unbound	LW11 [26], RW13 [31]	RW13 [31]	none
	full	small	unbound	LW12 [27]	LW12 [27]	none
	full	large	bound	OT12 [30]	OT12 [30]	none
	full	large	unbound	A14 [2]	**Ours**	**Ours**
Short-Cipher ABE	selective	large	unbound	ALP11 [6]	open[‡]	open
	full	large	unbound	A14 [2]	open[‡]	open
Short-Key ABE	selective	large	unbound	BGG+14 [8]	none	open
	full	large	unbound	open	**Ours**	open
(Bounded) ABE	selective	large	unbound	GPSW06 [21]	W11 [34]	AI09 [5]
	full	small	bound	LOS+10 [28]	LOS+10 [28]	**Ours**
	full	large	bound	OT10 [29], A14 [2]	OT10 [29], A14 [2]	**Ours**
Regular Languages	selective	small	-	W12 [35]	none	none
	full	large	-	A14 [2]	A14 [2]	**Ours**

[†] 'none' means that there was no previous work and it is subsumed by another system with stronger properties (*e.g.,* fully-secure). 'open' means that it remains an open problem. '-' means no defined property.

[‡] Short-cipher CP-ABE were given in [13, 24, 4, 11] but only for subclasses of span programs (AND, threshold).

w. We refer to [35, 2] for detailed definitions. By applying the generic conjunctive conversion and Theorem 9 to the KP-ABE-RL and CP-ABE-RL in [2], we obtain the first (fully-secure) DP-ABE-RL.

References

[1] J. Akinyele, C. Lehmann, M. Green, M. Pagano, Z. Peterson, A. Rubin, Self-Protecting Electronic Medical Records Using Attribute-Based Encryption. Cryptology ePrint Archive, Report 2010/565.

[2] N. Attrapadung. Dual System Encryption via Doubly Selective Security: Framework, Fully-secure Functional Encryption for Regular Languages, and More. In *Eurocrypt 2014*, *LNCS* 8441, pp. 557–577, 2014.

[3] N. Attrapadung. Fully Secure and Succinct Attribute Based Encryption for Circuits from Multi-linear Maps. Cryptology ePrint Archive: Report 2014/772.

[4] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. Panafieu, C. Rafols, Attribute-based encryption schemes with constant-size ciphertexts, In *Theoretical Computer Science* (422), pp. 15–38, 2012.

[5] N. Attrapadung, H. Imai. Dual-Policy Attribute Based Encryption. In *ACNS 2009*, *LNCS* 5536, pp. 168–185, 2009.

[6] N. Attrapadung, B. Libert, E. Panafieu. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts In *PKC 2011*, *LNCS* 6571, pp. 90–108, 2010.

[7] J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. In *IEEE Symposium on Security and Privacy 2007*, pp. 321–334, 2007.

[8] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, D. Vinayagamurthy. Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuits. In *Eurocrypt 2014*, *LNCS* 8441, pp. 533–556, 2014.

[9] D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt 2008*, *LNCS* 5350, pp. 455–470, 2008.

[10] D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC 2011*, *LNCS* 6597, pp. 253–273, 2011.

[11] C. Chen, J. Chen, H. W. Lim, Z. Zhang, D. Feng, S. Ling, H. Wang. Fully Secure Attribute-Based Systems with Short Ciphertexts/Signatures and Threshold Access Structures. In *CT-RSA 2013*, *LNCS* 7779, pp. 50–67, 2013.

[12] J. Coron, T. Lepoint, M. Tibouchi. Practical Multilinear Maps over the Integers. In *Crypto 2013*, *LNCS* 8042, pp. 476–493, 2013.

[13] K. Emura, A. Miyaji, A. Nomura, K. Omote, M. Soshi. A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length. In *ISPEC 2009*, *LNCS* 5451, pp. 13–23, 2009.

[14] S. Garg, C. Gentry, S. Halevi. Candidate multilinear maps from ideal lattices In *Eurocrypt 2013*, *LNCS* 7881, pp. 1–17, 2013.

[15] S. Garg, C. Gentry, S. Halevi, A. Sahai, B. Waters. Attribute-based encryption for circuits from multilinear maps. In *Crypto 2013*, *LNCS* 8042, pp. 479–499, 2013.

[16] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, B. Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits. In *FOCS 2013*, pp. 40–49, 2013.

[17] S. Garg, C. Gentry, S. Halevi, M. Zhandry. Fully Secure Attribute Based Encryption from Multilinear Maps. Cryptology ePrint Archive: Report 2014/622.

[18] S. Goldwasser, Y. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In *STOC 2013*, pp. 555–564, 2013.

[19] S. Goldwasser, Y. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich. How to run Turing machines on encrypted data. In *Crypto 2013*, *LNCS* 8042, pp. 536–553, 2013.

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

[21] V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS 2006*, pp. 89–98, 2006.

[22] V. Goyal, A. Jain, O. Pandey, A. Sahai. Bounded Ciphertext Policy Attribute Based Encryption. In *ICALP (2) 2008*, pp. 579–591, 2008.

[23] M. Hamburg. Spatial Encryption (Ph.D. Thesis). Cryptology ePrint Archive: Report 2011/389.

[24] J. Herranz, F. Laguillaumie, C. Ràfols. Constant-Size Ciphertexts in Threshold Attribute-Based Encryption. In *PKC 2010*, *LNCS* 6056, pp. 19–34, 2010.

[25] A. Lewko, B. Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010*, *LNCS* 5978, pp. 455–479, 2010.

[26] A. Lewko, B. Waters. Unbounded HIBE and Attribute-Based Encryption In *Eurocrypt 2011*, *LNCS* 6632, pp. 547–567, 2011.

[27] A. Lewko, B. Waters. New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In *Crypto 2012*, *LNCS*, 7417, pp. 180–198, 2012.

[28] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt 2010*, *LNCS* 6110, pp. 62–91, 2010.

[29] T. Okamoto, K. Takashima, Fully secure functional encryption with general relations from the decisional linear assumption.In *Crypto 2010*, *LNCS* 6223, pp. 191–208, 2010.

[30] T. Okamoto, K. Takashima, Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In *Asiacrypt 2012*, *LNCS* 7658, pp. 349–366, 2012.

[31] Y. Rouselakis, B. Waters Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM CCS 2013*, pp. 463–474, 2013.

[32] A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt 2005*, *LNCS* 3494, pp. 457–473, 2005.

[33] B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC 2011*, *LNCS* 6571, pp. 53–70, 2011.

[34] B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto 2009*, *LNCS* 5677, pp. 619–636, 2009.

[35] B. Waters. Functional Encryption for Regular Languages. In *Crypto 2012*, *LNCS* 7417, pp. 218–235, 2012.

[36] H. Wee. Dual System Encryption via Predicate Encodings. In *TCC 2014*, *LNCS* 8349, pp. 616-637, 2014.

A Our Ciphertext-policy Over Doubly Spatial Encryption

A.1 Definitions of KP-DSE, CP-DSE

We first recall the definition of affine spaces and KP-DSE [2] below. CP-DSE is defined as the dual of KP-DSE.

Affine Spaces. Let $N \in \mathbb{N}$. Let $M \in \mathbb{Z}_N^{n\times d}$ be a $n \times d$ matrix whose columns are all linearly independent and $\boldsymbol{c} \in \mathbb{Z}_N^n$ be a (horizontal) vector. We define an affine space by $V(M, \boldsymbol{c}) = \left\{ \boldsymbol{w}M^\top + \boldsymbol{c} \,\middle|\, \boldsymbol{w} \in \mathbb{Z}_N^d \right\}$. In what follows, we will also use a shorter representation using affine matrices. Let $\mathsf{Aff}(\mathbb{Z}_N^d) = \left\{ (1, \boldsymbol{w}) \,\middle|\, \boldsymbol{w} \in \mathbb{Z}_N^d \right\}$. We call $X = \left(\begin{smallmatrix} 1 & \boldsymbol{0}_d \\ \boldsymbol{c}^\top & M \end{smallmatrix}\right)$ an affine matrix for the affine space $V(M, \boldsymbol{c})$ and define $\mathsf{AffSp}(X) = \{ (1, \boldsymbol{w}) \mid \boldsymbol{w} \in V(M, \boldsymbol{c}) \}$. Hence, we have $\mathsf{AffSp}(X) = \left\{ \boldsymbol{v}X^\top \,\middle|\, \boldsymbol{v} \in \mathsf{Aff}(\mathbb{Z}_N^d) \right\}$. Denote the set of all affine matrices with dimension $(n+1) \times (d+1)$ by $\mathsf{AffM}(\mathbb{Z}_N^{n\times d}) = \left\{ \left(\begin{smallmatrix} 1 & \boldsymbol{0}_d \\ \boldsymbol{c}^\top & M \end{smallmatrix}\right) \,\middle|\, M \in \mathbb{Z}_N^{n\times d}, \mathrm{rank}(M) = d, \boldsymbol{c} \in \mathbb{Z}_N^n \right\}$. If $\mathsf{AffSp}(T) \subseteq \mathsf{AffSp}(X)$, we have an efficient algorithm to compute an affine matrix D such that $T = XD$, see [9, 23].

Doubly Spatial Encryption. Doubly-spatial predicate [23] is indexed by the full dimension of spaces, denoted by n. In doubly-spatial encryption [23], we have a key and a ciphertext associated to affine spaces $X \in \mathsf{AffM}(\mathbb{Z}_N^{n\times *})$ and $Y \in \mathsf{AffM}(\mathbb{Z}_N^{n\times *})$ respectively. The relation is defined by $R^{\mathsf{DS}}(X, Y) = 1$ if and only if $\mathsf{AffSp}(X) \cap \mathsf{AffSp}(Y) \neq \emptyset$, *i.e.,* both affine spaces are intersected.

Linear Secret Sharing (LSS). We briefly review some facts about LSS. Consider a matrix $A \in \mathbb{Z}_N^{m\times k}$. Consider a set $S \subseteq [1, m]$. Let A_S be the sub-matrix of A that takes exactly all the rows in S. We say that A accepts S if and only if $(1, \boldsymbol{0}) \in \mathsf{RowSp}(A_S)$. A corresponds to an LSS scheme for m parties and the authorized sets are those which A accepts. Such an LSS scheme is implemented as follows. To share $\alpha_1 \in \mathbb{Z}_N$, the dealer chooses $\alpha_2, \dots, \alpha_k \xleftarrow{\$} \mathbb{Z}_N$, and makes a share $A_i\boldsymbol{\alpha}^\top$ to the i-th party. Parties in an accepted set S can reconstruct α_1 since $(1, \boldsymbol{0}) \in \mathsf{RowSp}(A_S)$ hence $\{\mu_j\}_{j\in S}$ such that $\sum_{j\in S} \mu_j A_j = (1, \boldsymbol{0})$ can be obtained and we observe that $\sum_{j\in S} \mu_j A_j \alpha^\top = (1, \boldsymbol{0})\alpha^\top = \alpha_1$. *Key-Policy over Doubly Spatial Encryption* (KP-DSE) is defined via the following definition.

Definition 2 (Key-Policy over Primitives [2])**.** Consider a predicate $R : \mathbb{X} \times \mathbb{Y}$. Consider an *access matrix* $A \in \mathbb{Z}_N^{m\times k}$ for a linear secret sharing scheme. We associate a key attribute $X^{(i)} \in \mathbb{X}$ to row i of A and call $\mathbb{A} = \left(A; X^{(1)}, \dots, X^{(m)}\right)$ an *access structure over* $\mathbb{X}$. Let $\mathcal{A}_\mathbb{X}$ be the universe of considered access structures over $\mathbb{X}$. For a set $\Omega = \{Y^{(1)}, \dots, Y^{(t)}\} \subseteq \mathbb{Y}$, we define $Q_\Omega = \{i \in [1, m] \mid \exists Y^{(j)} \in \Omega \text{ s.t. } R(X^{(i)}, Y^{(j)}) = 1\}$. We define a new relation $\tilde{R}$ over $\mathcal{A}_\mathbb{X} \times 2^\mathbb{Y}$ as $\tilde{R}\big(\mathbb{A}, S\big) = 1$ if and only if A accepts Q_Ω. We call $\tilde{R}$ the predicate of key-policy over R.

A.2 Recapturing the KP-DSE scheme of [2]

For self-containment, we recall the pair encoding scheme for KP-DSE of [2] as follows. We refer the correctness to [2].

Pair Encoding Scheme 1: $\mathsf{P}_{\mathsf{KPDSE}}$ for KP-DSE of [2]

$\mathsf{Param}(n) \to n+5$. Denote $\boldsymbol{h} = (\boldsymbol{t}, \phi_1, \phi_2, \phi_3, \eta)$, where $\boldsymbol{t} = (h_0, h_1, \dots, h_n)$.

$$\begin{array}{ll} & \text{For } \mathbb{A} = (A; X^{(1)}, \dots, X^{(m)}) \text{ where } A \in \mathbb{Z}_N^{m\times k} \text{ and } X^{(i)} \in \mathsf{AffM}(\mathbb{Z}_N^{n\times d_i}) \\ \mathsf{Enc1}(\mathbb{A}) & \to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = (k_1, k_2, k_3, \{k_{4,i}, k_{5,i}, \boldsymbol{k}_{6,i}\}_{i\in[1,m]}) : \\ & \left\{ \begin{array}{lll} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_{4,i} = A_i \boldsymbol{v}^\top + r_i\phi_3, & k_{5,i} = r_i, & \boldsymbol{k}_{6,i} = r_i \boldsymbol{t} X^{(i)} \end{array} \right\} \\ & \text{where } v_1 := r\phi_2 \text{ and } \boldsymbol{r} = (r, r_1, \dots, r_m, u, v_2, \dots, v_k),\ \boldsymbol{v} := (v_1, v_2, \dots, v_k). \end{array}$$

$$\begin{array}{ll} & \text{For } \Omega = \{Y^{(1)}, \dots, Y^{(t)}\} \text{ where } Y^{(j)} \in \mathsf{AffM}(\mathbb{Z}_N^{n\times f_j}) \\ \mathsf{Enc2}(\Omega) & \to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = (c_1, c_2, c_3, c_4, \{\boldsymbol{c}_{5,j}, c_{6,j}\}_{j\in[1,t]}) : \\ & \left\{ \begin{array}{lll} c_1 = s_0, & c_2 = s_0\eta, & c_3 = s_0\phi_1 + w\phi_2, \\ c_4 = w, & \boldsymbol{c}_{5,j} = (w\phi_3, \boldsymbol{0}) + s_j \boldsymbol{t} Y^{(j)}, & c_{6,j} = s_j \end{array} \right\} \\ & \text{where } \boldsymbol{s} = (s_0, s_1, \dots, s_t, w). \end{array}$$

A.3 Our New CP-DSE with Tighter Reduction

Let $\mathsf{P}_{\mathsf{KPDSE}}$ denote the pair encoding for KP-DSE of [2]. We describe new CP-DSE, obtained as $\mathcal{EC}(\mathsf{P}_{\mathsf{KPDSE}})$, below. We recall that $\mathcal{EC}$ is the extended dual conversion given in §5.

The non-barred variables depict the elements from $\mathsf{P}_{\mathsf{KPDSE}}$, while the barred ones are introduced via the extended conversion $\mathcal{EC}$. Its correctness thus follows from those of $\mathsf{P}_{\mathsf{KPDSE}}$ and $\mathcal{EC}$, and we omit it here.

Pair Encoding Scheme 2: $\mathcal{EC}(\mathsf{P}_{\mathsf{KPDSE}})$ for Our New CP-DSE

$\mathsf{Param}(n) \to n+7$. Denote $\bar{\boldsymbol{h}} = (\boldsymbol{t}, \phi_1, \phi_2, \phi_3, \eta, \bar{\phi}, \bar{\eta})$, where $\boldsymbol{t} = (h_0, h_1, \dots, h_n)$.

$$\begin{array}{ll} & \text{For } \Omega = \{Y^{(1)}, \dots, Y^{(t)}\} \text{ where } Y^{(j)} \in \mathsf{AffM}(\mathbb{Z}_N^{n\times f_j}) \\ \mathsf{Enc1}(\Omega) & \to \bar{\boldsymbol{k}}(\bar{\alpha}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{h}}) = (c_1, c_2, c_3, c_4, \{\boldsymbol{c}_{5,j}, c_{6,j}\}_{j\in[1,t]}, \bar{k}_1, \bar{k}_2) : \\ & \left\{ \begin{array}{lll} c_1 = s_0, & c_2 = s_0\eta, & c_3 = s_0\phi_1 + w\phi_2, \\ c_4 = w, & \boldsymbol{c}_{5,j} = (w\phi_3, \boldsymbol{0}) + s_j \boldsymbol{t} Y^{(j)}, & c_{6,j} = s_j \\ \bar{k}_1 = \bar{\alpha} + s_0\bar{\phi} + \bar{u}\bar{\eta}, & \bar{k}_2 = \bar{u} & \end{array} \right\} \\ & \text{where } \bar{\boldsymbol{r}} = (s_0, s_1, \dots, s_t, w, \bar{u}). \end{array}$$

$$\begin{array}{ll} & \text{For } \mathbb{A} = (A; X^{(1)}, \dots, X^{(m)}) \text{ where } A \in \mathbb{Z}_N^{m\times k} \text{ and } X^{(i)} \in \mathsf{AffM}(\mathbb{Z}_N^{n\times d_i}) \\ \mathsf{Enc2}(\mathbb{A}) & \to \bar{\boldsymbol{c}}(\bar{\boldsymbol{s}}, \bar{\boldsymbol{h}}) = (k_1, k_2, k_3, \{k_{4,i}, k_{5,i}, \boldsymbol{k}_{6,i}\}_{i\in[1,m]}\bar{c}_1, \bar{c}_2) : \\ & \left\{ \begin{array}{lll} k_1 = \bar{s}_0\bar{\phi} + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_{4,i} = A_i \boldsymbol{v}^\top + r_i\phi_3, & k_{5,i} = r_i, & \boldsymbol{k}_{6,i} = r_i \boldsymbol{t} X^{(i)} \\ \bar{c}_1 = \bar{s}_0, & \bar{c}_2 = \bar{s}_0\bar{\eta} & \end{array} \right\} \\ & \text{where } v_1 := r\phi_2 \text{ and } \bar{\boldsymbol{s}} = (\bar{s}_0, r, r_1, \dots, r_m, u, v_2, \dots, v_k),\ \boldsymbol{v} := (v_1, v_2, \dots, v_k). \end{array}$$

We will prove the $(1, \mathsf{poly})$-SMH security of our CP-DSE in §A.3 under a new assumption, called EDHE4-Dual, that is similar to the EDHE4 Assumption used for the $(1,1)$-CMH security for the KP-DSE of [2]. The only modification consists of two elements that we write in the the box in Definition 3 below. We describe it in Definition 3.

Definition 3 ((n,m,k)-EDHE4-Dual Assumption)**.** The (n,m,k)-Expanded Diffie-Hellman Exponent Assumption-4-Dual in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$}$

$\mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, x, c, b_1, \ldots, b_m, \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$. Suppose that an adversary is given $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of $g, g^c, \boxed{g^{c/z}}$, $\forall_{j\in[1,k]}\ g^{a^{n+1}x^j}$ and

$$\begin{aligned}
\forall_{i\in[1,n],\ j\in[1,k],\ \iota\in[1,m]} &\quad g^{a^i x^j/b_\iota^2}, g^{cb_\iota}, g^{x^j}, g^{a^i x^j b_\iota} \\
\forall_{j\in[1,k],\ \iota,\iota'\in[1,m],\iota\neq\iota'} &\quad g^{a^{n+1}x^j cb_\iota/b_{\iota'}} \\
\forall_{i\in[1,n],\ j\in[1,k],\ \iota,\iota'\in[1,m],\iota\neq\iota'} &\quad g^{a^i x^j cb_\iota/b_{\iota'}^2} \\
\forall_{i\in[1,2n],\ j\in[1,2k],\ \iota,\iota'\in[1,m],\ (i,j,\iota)\neq(n+1,k+1,\iota')} &\quad g^{a^i x^j b_\iota/b_{\iota'}^2}
\end{aligned}$$

and the other generators g_1, g_3. The assumption states that it is hard for any polynomial-time adversary to distinguish whether $\boxed{T = g^{a^{n+1}x^{k+1}z}}$ or $T \xleftarrow{\$} \mathbb{G}_p$.

We are ready to describe the security theorem for our CP-DSE encoding.

Theorem 11. *The pair encoding $\mathcal{EC}(\mathsf{P}_{\mathsf{KPDSE}})$ is $(1, \mathsf{poly})$-SMH-secure under the (n, m, k)-$\mathsf{EDHE4\text{-}Dual}$ assumption with tight reduction, where n is the full dimension of spaces in the ciphertext query $\mathbb{A}$ and (m, k) is the matrix dimension of the access matrix A of $\mathbb{A}$.*

We also obtain Corollary 12 below which follows immediately from Theorem 11, Proposition 1, and the application of Corollary 7 to the $(1, 1)$-SMH of the KP-DSE of [2], which was proved under the $\mathsf{EDHE3}$ Assumption.

Corollary 12. *For any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5$, whose running times are the same as $\mathcal{A}$ plus some polynomial times, such that for any λ,*

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}(\mathcal{EC}(\mathsf{P}_{\mathsf{KPDSE}}))}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SD1}}(\lambda) + (2q_1 + 3)\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{SD2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SD3}}(\lambda) \\
+ q_1\mathsf{Adv}_{\mathcal{B}_4}^{(n,t)\text{-}\mathsf{EDHE3}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_5}^{(n,m,k)\text{-}\mathsf{EDHE4\text{-}Dual}}(\lambda),
\end{aligned}$$

where q_1 is the number of queries in phase 1, n is the full dimension of spaces, t is the maximum number of spaces in Ω among pre-challenge key queries, and (m, k) is the matrix dimension of the access matrix A in $\mathbb{A}$ for the challenge ciphertext query.

A.4 Security Proof for Our CP-DSE Encoding

The proof of Theorem 11 mostly follows co-selective security proof of the original pair encoding scheme $\mathsf{P}_{\mathsf{KPDSE}}$ for KP-DSE of [2]. We briefly describe the difference. Most of the original variables are simulated in exactly the same manner. The unmodified variables comprise $\phi_2, \phi_3, \boldsymbol{t}$ (for parameter variables), $r, r_i, \boldsymbol{v}$ (for keys in KP-DSE, hence for ciphertexts in CP-DSE), and $s_1, \ldots, s_t, w$ (for ciphertexts in KP-DSE, hence for keys in CP-DSE). The only previous variables that will be simulated differently from [2] are ϕ_1, η, u, s_0. Moreover, we simulate also new variables (those barred ones), which consist of $\bar{\phi}, \bar{\eta}, \bar{u}, \bar{s}_0$.

For clarity and conciseness, we mostly refer to the KP-DSE proof (the proof of Theorem 12 in the full version of [2]) and show only the difference here in the proof sketch below.

Proof of Theorem 11 (Sketch). Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the selective master-key hiding game against the encoding scheme $\mathcal{EC}(\mathsf{P}_{\mathsf{KPDSE}})$. We construct a simulator $\mathcal{B}$ that solves the (n,m,k)-$\mathsf{EDHE4\text{-}Dual}$ Assumption that uses $\mathcal{A}$ as follows.

Simulating $\mathcal{O}^1$ for Ciphertext Query. The game begins with $\mathcal{A}$ making a ciphertext query for $\mathbb{A} = (A, \{X^{(i)}\}_{i\in[1,m]})$ where $X^{(i)} \in \mathsf{AffM}(\mathbb{Z}_p^{n\times d_i})$, $A \in \mathbb{Z}_p^{m\times k}$. $\mathcal{B}$ takes the (n,m,k)-$\mathsf{EDHE4\text{-}Dual}$ challenge $(\boldsymbol{D}, T)$, where $T = g^{\tau + a^{n+1}x^{k+1}z}$. Its task is to guess if $\tau = 0$ or τ is random in $\mathbb{Z}_p$.

(Programming Parameters). $\mathcal{B}$ first compute for $i \in [1,m]$ an affine matrix $K_i \in \mathsf{AffM}(\mathbb{Z}_p^{(n-d_i)\times n})$ such that $K_i X^{(i)} = \begin{pmatrix} 1 & \mathbf{0}_{d_i} \\ \mathbf{0}_{n-d_i}^\top & \mathbf{0} \end{pmatrix} \in \mathsf{AffM}(\mathbb{Z}_p^{(n-d_i)\times d_i})$, where here $\mathbf{0} \in \mathbb{Z}_p^{(n-d_i)\times d_i}$. This can be done as described in [23, 2]. We denote the following vectors to be used throughout the proof.

$$\begin{aligned} \forall_{i\in[1,m]} \qquad \boldsymbol{a}^{(i)} &= (a, a^2, \ldots, a^{n-d_i}) \in \mathbb{Z}_p^{n-d_i}, & \boldsymbol{\sigma}^{(i)} &= (a^n, a^{n-1}, \ldots, a^{d_i+1}) \in \mathbb{Z}_p^{n-d_i} \\ \boldsymbol{x} &= (x, x^2, \ldots, x^k) \in \mathbb{Z}_p^k, & \boldsymbol{\chi} &= (x^k, x^{k-1}, \ldots, x) \in \mathbb{Z}_p^k. \end{aligned}$$

$\mathcal{B}$ computes the parameter by choosing $\boldsymbol{t'} \xleftarrow{\$} \mathbb{Z}_p^{n+1}, \phi'_1, \phi'_2, \phi'_3, \bar{\phi}', \eta, \bar{\eta} \xleftarrow{\$} \mathbb{Z}_p$ then implicitly sets

$$\bar{\phi} = xz + \bar{\phi}' \qquad \phi_1 = x + \phi'_1, \qquad \phi_2 = a^{n+1}x + \phi'_2, \qquad \bar{\eta} = z$$

$$\boldsymbol{t} = \sum_{i\in[1,m]} (A_i \boldsymbol{x}^\top)(0, \frac{\boldsymbol{a}^{(i)}}{b_i^2})K_i + \boldsymbol{t}', \qquad \phi_3 = \sum_{i\in[1,m]} -(A_i \boldsymbol{x}^\top)\frac{a^{n+1}}{b_i} + \phi'_3.$$

We note that $g^{\boldsymbol{t}}, g^{\phi_1}, g^{\phi_2}, g^{\phi_3}$ can be computed from $\boldsymbol{D}$ while g^{η} is also trivially computable, but $\mathcal{B}$ cannot compute $g^{\bar{\phi}}, g^{\bar{\eta}}$ since it does not possess g^{xz}, g^{z}.

(Programming Randomness in Ciphertext). We now describe how $\mathcal{B}$ produce a ciphertext for $\mathbb{A}$. $\mathcal{B}$ chooses $u \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{B}$ then implicitly sets

$$\bar{s}_0 = -c/z, \qquad r = c, \qquad \forall_{i\in[1,m]}\ r_i = cb_i, \qquad \boldsymbol{v} = a^{n+1}c\boldsymbol{x} + (\phi'_2 c, \mathbf{0}),$$

where v_1 correctly obeys $v_1 = r\phi_2$. For clarity, we will re-randomize the randomness variables later.

(Cancellation and Simulation for Ciphertext). From the above (implicit) definition, the ciphertext is well defined. It can be computed due to the cancelation of unknown elements as follows. The cancellations of $g^{x^j c\boldsymbol{a}^{(i)}/b_i}$ terms in $g^{r_i \boldsymbol{t} X^{(i)}}$ and of $g^{cx^j a^{n+1}}$ in $g^{A_i \boldsymbol{v}^\top + r_i\phi_3 + u\eta}$ are exactly the same as in the KP-DSE proof. The only new cancelation is for canceling g^{cx} in $g^{\bar{s}_0\bar{\phi} + r\phi_1 + u\eta}$, which is straightforward. The ciphertext is then computable from remaining terms which consist of only known elements from $\boldsymbol{D}$. This follows exactly the KP-DSE proof except that we also have new elements where we compute:

$$g^{\bar{s}_0\bar{\phi} + r\phi_1 + u\eta} = (g^{-c/z})^{\bar{\phi}'}(g^c)^{\phi'_1} g^{u\eta}, \qquad g^{\bar{s}_0} = g^{-c/z}, \qquad g^{\bar{s}_0\bar{\eta}} = g^{-c}.$$

We also note that u, η is different from the KP-DSE proof, where here $\mathcal{B}$ chose u, η by itself.

(Re-randomizing Ciphertext). The simulated ciphertext is not perfectly distributed yet since their randomness variables are still correlated. We re-randomize every variable y in $\bar{\boldsymbol{s}}$ besides $\bar{s}_0$ (and u, since it is already randomly chosen), so that it is independent from $\bar{s}_0$.

We re-randomize each to $y'' = y + y'$, where $\mathcal{B}$ chooses $y' \xleftarrow{\$} \mathbb{Z}_p$. This can be done since $\mathcal{B}$ possesses $g, g^{\boldsymbol{t}}, g^{\phi_1}, g^{\phi_2}, g^{\phi_3}, g^{\eta}$.

Simulating $\mathcal{O}^2$ for Key Query. $\mathcal{A}$ makes a key query for $\Omega = (Y^{(1)}, \ldots, Y^{(t)})$. Let $S \subseteq [1, m]$ be the set of all i such that there exists $j \in [1, t]$ where $R^{\mathsf{DS}}(X^{(i)}, Y^{(j)}) = 1$. Let A_S be the sub-matrix of A that contains only rows in S. From the restriction that $R_{\mathsf{KP}(DS)}(\mathbb{A}, \Omega) = 0$ and from the definition of LSSS, we must have that $(1, \mathbf{0}) \notin \mathsf{RowSp}(A_S)$. Using Proposition 39 in [2], we can find $\boldsymbol{\nu} \in \mathbb{Z}_p^k$ such that $\nu_1 = -1$ and $A_i \boldsymbol{\nu}^\top = 0$ for all $i \in S$.

(Programming Randomness in Key). To construct a key, $\mathcal{B}$ first implicitly sets

$$\bar{\alpha} = \tau, \qquad \bar{u} = 0, \qquad s_0 = x^k a^{n+1}, \qquad w = \boldsymbol{\chi}\boldsymbol{\nu}^\top.$$

Then, for $j \in [1, t]$, $\mathcal{B}$ constructs the corresponding key element as follows. Consider $i \notin S$. We have that $R^{\mathsf{DS}}(X^{(i)}, Y^{(j)}) = 0$. We then define $\boldsymbol{w}^{(j,i)} \in \mathbb{Z}_p^{n-d_i}$ as in the KP-DSE proof. $\mathcal{B}$ then implicitly sets $s_j = (\boldsymbol{\chi}\boldsymbol{\nu}^\top)(\sum_{i \in [1,m] \text{ s.t. } i \notin S} b_i \boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top)$, as in the KP-DSE proof.

(Cancellation and Simulation for Key). From these (implicit) definitions, the key is well defined. It can be computed due to the cancelation of unknown elements as follows. Cancelation of $g^{a^{n+1} x^{k+1}/b_i}$ for all $i \in [1, m]$ in $g^{(w\phi_3, \mathbf{0}) + s_j \boldsymbol{t} Y^{(j)}}$ is exactly the same as the KP-DSE proof, while cancelation of $g^{x^{k+1} a^{n+1}}$ in $g^{s_0 \phi_1 + w \phi_2}$ is also similar (even s_0, ϕ_1 is simulated differently here). Due to these cancelations, the key can be computed from available terms from $\boldsymbol{D}$. This follows exactly the KP-DSE proof except that we also have new elements where we compute

$$g^{\bar{k}_1} = T \cdot (g^{s_0})^{\bar{\phi}'} = g^{\tau + (x^k a^{n+1})(xz)} g^{s_0 \bar{\phi}'} = g^{\bar{\alpha} + s_0 \bar{\phi} + \bar{u}\bar{\eta}}, \qquad g^{\bar{k}_2} = 1 = g^{\bar{u}}.$$

Also note a difference that $\mathcal{B}$ chose η by itself.

(Re-randomizing Key). The simulated key is not perfectly distributed yet since their randomness are still correlated. $\mathcal{B}$ re-randomizes the key by choosing $s'_0, s'_1, \ldots, s'_t, w', \bar{u}' \xleftarrow{\$} \mathbb{Z}_p$ and computing a key that has new randomness $s''_i = s_i + s'_i$ for $i \in [0, t]$, $w'' = w + w'$, and $\bar{u}'' = \bar{u} + (c/z)\bar{u}' - xs'$. This can be done since $\mathcal{B}$ possesses $g, g^{\boldsymbol{t}}, g^{\phi_1}, g^{\phi_2}, g^{\phi_3}, g^{\eta}$. We note a crucial point that although $\mathcal{B}$ does not possess $g^{\bar{\phi}}, g^{\bar{\eta}}$, it can re-randomize the corresponding elements as

$$g^{\bar{\alpha} + s''_0 \bar{\phi} + \bar{u}'' \bar{\eta}} = g^{\bar{\alpha} + s_0 \bar{\phi} + \bar{u}\bar{\eta}} (g^{\bar{\phi}'})^{s'_0} (g^{c})^{u'}, \qquad g^{\bar{u}''} = g^{\bar{u}} (g^{c/z})^{\bar{u}'} (g^{x})^{-s'_0},$$

where we essentially use the *randomizer technique* from [27, 2], which allows us to simulate polynomially many keys. This is actually the reason why we need one more element each for a key and a ciphertext for the extended dual conversion.

Guess. $\mathcal{A}$ will eventually output a guess if $\bar{\alpha} = 0$ or random, our simulator $\mathcal{B}$ just outputs the guess that $\tau = 0$ and τ is random respectively. Since $\bar{\alpha} = \tau$, the advantage of $\mathcal{B}$ winning the (n, m, k)-$\mathsf{EDHE4\text{-}Dual}$ assumption is exactly the same as the advantage of $\mathcal{A}$ winning the selective master-key hiding game. □