Unbounded Dynamic Predicate Compositions in ABE from Standard Assumptions

Junichi Tomida[1] and Nuttapong Attrapadung[2]

[1] NTT Corporation, Japan
junichi.tomida.vw@hco.ntt.co.jp
[2] AIST, Japan
n.attrapadung@aist.go.jp

Abstract. At Eurocrypt'19, Attrapadung presented several transformations that dynamically compose a set of attribute-based encryption (ABE) schemes for simpler predicates into a new ABE scheme for more expressive predicates. Due to the powerful unbounded and modular nature of his compositions, many new ABE schemes can be obtained in a systematic manner (including those that resolved some open problems at the time). However, his approach heavily relies on so-called q-type assumptions, which are not standard. Devising such powerful compositions from standard assumptions was left as an important open problem. In this paper, we present a new framework for constructing ABE schemes that allow unbounded and dynamic predicate compositions among them, and show that the adaptive security of these composed ABE will be preserved by relying only on the standard matrix Diffie-Hellman (MDDH) assumption. This thus resolves the open problem posed by Attrapadung. Our framework significantly expands an area of ABEs that are realizable from standard assumptions. This includes the following adaptively secure *large-universe* ABEs for Boolean formulae under MDDH.

- The first completely unbounded monotone key-policy (KP)/ciphertext-policy (CP) ABE. Previously, such ABE has been only recently proposed, but only for the KP and *small-universe* flavor (Kowalczyk and Wee, Eurocrypt'19).
- The first completely unbounded non-monotone KP/CP-ABE. Especially, our ABEs support a new type of non-monotonicity that subsumes previous two types of non-monotonicity, namely, by Ostrovsky *et al.* (CCS'07) and by Okamoto and Takashima (CRYPTO'10).
- The first non-monotone KP and CP-ABE with constant-size ciphertexts and secret keys, respectively.
- The first monotone KP and CP-ABE with constant-size secret keys and ciphertexts, respectively.

Keywords: Attribute-based encryption, predicate compositions, standard assumptions, completely unbounded ABE, non-monotone ABE, succinct ABE

Table of Contents

1 Introduction

Attribute-based encryption (ABE) is a generalized form of public-key encryption that allows fine-grained access control over encrypted data [22, 32]. In a broader sense of ABE, each scheme specifies a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are ciphertext and secret-key attribute universes, respectively. All users can encrypt a message with an arbitrary attribute $x \in \mathcal{X}$ using public information. An owner of a master secret key can generate a secret key for an arbitrary attribute $y \in \mathcal{Y}$. A ciphertext for attribute x is decryptable with a secret key for attribute y if and only if x and y satisfy the predicate P, i.e., $\mathsf{P}(x,y) = 1$. This is in contrast to the traditional public-key encryption, in which only one legitimate user can decrypt a ciphertext.

One of central research topics in ABE is to explore what kind of predicates for which ABE can be realized. This is important in practice since if one attempts to realize an access control system based on ABE, the underlying predicate must be able to express all decryption conditions that appear in the system. A line of works has shown that we can realize ABE for various predicates: ABE for span programs, (non-)deterministic finite automata, polynomial-sized circuits, and so on [4, 12, 21, 22, 24, 30, 32, 36]. These works directly construct ABE schemes for targeting predicates. In contrast, there is also another approach to construct ABE schemes for more expressive new predicates by transformations and combinations of known predicates [5, 6, 8, 11]. The state of the art on this approach is the work by Attrapadung [8], who proposed a framework for dynamic predicate compositions and introduced new ABE schemes such as ABE for key-policy (KP)/ciphertext-policy (CP) augmentation over predicate sets, nested-policy ABE, and mixed-policy ABE. The salient feature of these ABE schemes is that they allow *unbounded* and *dynamic* predicate compositions, that is, they do not impose any restriction on the size and structure of composition policy. This is in contrast to previous works [5, 6, 11], which allow only *static* (*i.e.,* a-priori fixed) compositions. He also showed that his framework captures predicates that are known but whose adaptively secure ABE instance was still open such as the predicate for completely unbounded non-monotone ABE.

His framework takes a modular approach to construct a new predicate with a corresponding pair encoding scheme (PES), which is an encoding system that yields a concise expression of an ABE scheme [6]. It is shown in [8] that a nested application of three transformations of predicates, namely, direct sum, dual transformation, and KP augmentation over a single predicate (we call it just KP augmentation in what follows), is sufficiently powerful to obtain expressive predicates, such as the predicates for KP/CP augmentation over predicate sets, nested-policy ABE, and completely unbounded non-monotone ABE. He also demonstrates the transformations of PESs that correspond to the three transformations of the predicates. Hence, starting from known predicates and corresponding PESs, one can obtain a new transformed predicate along with its PES. Additionally, all PESs obtained in his framework can be used to instantiate a secure ABE scheme.

A crucial fact that his framework relies on is that the transformations of PESs preserve the symbolic property, introduced by Agrawal and Chase [3]. That is, he proved that all transformed PESs in his framework satisfy the symbolic property if the starting PESs satisfy the symbolic property. Agrawal and Chase showed that an ABE scheme induced by a PES with the symbolic property is adaptively secure under the q-ratio assumption [3]. Thus, we can use known predicates that have a PES with the symbolic property to construct a new expressive predicate and the corresponding PES, which results in a secure ABE scheme.

One drawback of his framework is the necessity of the q-ratio assumption, which is one of so-called q-type assumptions. The q-ratio assumption is parame-trized with two parameters d_1 and d_2 and becomes stronger as they grow. We require that the q-ratio assumption holds with respect to sufficiently large d_1 and d_2 to assure the security of most ABE schemes because these parameters depend on adversary's behavior. However, the q-ratio assumption is a new complex assumption and thus not well-understood. Hence, it is desirable if we can transform PESs and instantiate an ABE scheme from a transformed PES under well-understood standard assumptions like the matrix Diffie-Hellman assumption, instead of q-type assumptions. The realization of such a framework yields many important new ABEs from standard assumptions but has been left as an open problem by Attrapadung [8].

Table 1. Comparison among frameworks that compose multiple predicates over ABE.

Framework	Composition type	Comp. class	Input primitive	Assumption
ABS17 [5]	Static	Boolean formulae	Predicate encodings (info.-theoretic)	MDDH
Att19 [8]	Unbounded, Dynamic	SP, BP, DFA	Pair encodings with symbolic security	q-ratio
This work	Unbounded, Dynamic	Boolean formulae	Pair encodings with info.-theoretic security or with Key-Encoding Indistinguishability	MDDH

Note: SP, BP, DFA stand for span programs, branching programs, deterministic finite automata, respectively.

1.1 Our Contributions

New Framework. We give an affirmative answer to the problem and present a new framework for transforming predicates and constructing ABE schemes on prime-order bilinear groups, which relies on only the standard matrix Diffie-Hellman (MDDH) assumption. Following [8], our framework also composes a new predicate by combining three essential transformations, namely, the direct sum, dual transformation, and KP augmentation. Nested applications of these transformations yield various expressive predicates and ABE schemes. Our framework introduces a new property on PESs that satisfies the two requirements under the MDDH assumption: the preservation of the property in the transformations and the induction of the adaptive security of the resulting ABE scheme.

Note that there are two differences between our framework and that by Attrapadung [8] (we provide a comparison among composition frameworks in Table 1). First, our KP augmentation is done with Boolean formulae, whereas that by Attrapadung is augmentation with span programs, branching programs, and deterministic finite automata (realizing them from standard assumptions is an interesting open problem). Second, starting predicates need to have a PES with a certain information-theoretic property, whereas those in his framework only require a PES with the symbolic property. Note that the latter may be attainable by larger classes of predicates (but the symbolic property would require q-type assumptions). Nevertheless, our framework is still sufficiently powerful to realize many ABE schemes of which instantiations under the standard assumptions have remained open before our work.

New Instantiations. Via our new framework, we obtain the following ABE instantiations for important specific predicates. We emphasize that all the instantiations are *large-universe* constructions, which have a super-poly size attribute domain. Their comparisons to previous schemes are given in Tables 2 to 5.

1. The first adaptively secure completely unbounded KP/CP-ABE for monotone Boolean formulae under MDDH.[3] Previously, such an adaptively secure KP/CP-ABE relies on either q-type assumptions [3, 7, 8] or the one-use restriction (each attribute is usable at most once in a policy) [15, 29]. Note that the recent unbounded KP-ABE with multi-use by Kowalczyk and Wee [25, §A] is a *small-universe* construction, i.e., the attribute domain size is (a priori unbounded) polynomial.
2. The first adaptively secure completely unbounded KP/CP-ABE for *non-monotone* Boolean formulae under MDDH. Furthermore, our ABE schemes support a new type of non-monotonicity that conflates the two types of existing non-monotonicity by Ostrovsky, Sahai, and Waters (OSW) [30] and by Okamoto and Takashima (OT) [28]. In other words, both OSW-non-monotone ABE and OT-non-monotone ABE can be captured as a special case of our non-monotone ABE. Previously, an adaptively secure unbounded ABE for non-monotone formulae is either the OSW type and based on q-type assumption [8] or the OT type with the one-use restriction [29].
3. The first adaptively secure KP/CP-ABE with constant-size ciphertexts/secret keys under MDDH for (OSW-non-)monotone Boolean formulae, respectively.

[3] To be more precise, we describe some terms. *Unbounded ABE* [27] refers to schemes that have no bounds on the sizes of attribute sets (inputs to a Boolean formula) and policies (Boolean formulae). *Multi-use* refers to the property that any attribute can be used arbitrarily many times in one policy. *Completely unbounded ABE* refers to unbounded *large-universe* ABE with multi-use (see *e.g.,* [8]).

Table 2. Comparison among *unbounded* ABE schemes.

References	Large universe	Adaptive security	Multi-use	Static assumption	Without RO	Non-monotonicity	Prime-order	KP/CP
LW11 [27]	✓		✓	✓	✓			KP
OT12 [29]	✓	✓		✓	✓	✓(OT)	✓	KP, CP
RW13 [31]	✓		✓		✓		✓	KP, CP
YAHK14 [38]	✓		✓		✓	✓(OSW)	✓	KP, CP
Att14 [6]	✓	✓	✓		✓			KP
AY15 [11]	✓	✓	✓		✓			CP
Att16 [7]	✓	✓	✓		✓		✓	KP, CP
AC17a [3]	✓	✓	✓		✓		✓	KP, CP
AC17b [2]	✓	✓		✓			✓	KP, CP
CGKW18 [15]		✓		✓	✓		✓	KP, CP
KW19 [25]		✓	✓	✓	✓		✓	KP
Att19 [8]	✓	✓	✓		✓	✓(OSW)	✓	KP, CP
TKN19 [34]	✓	✓	✓	✓		✓(OT)	✓	KP, CP
Ours 1	✓	✓	✓	✓	✓		✓	KP, CP
Ours 2	✓	✓	✓	✓	✓	✓(OSWOT)	✓	KP, CP

Note: KP, CP is for key-policy, ciphertext-policy. RO is for random oracles. We consider three types of non-monotone ABE: OT-type (Okamoto-Takashima [29]), OSW-type (Ostrovsky-Sahai-Waters [30]), and a new unified type (OSWOT) (see §6).

Table 3. Closer comparison among *adaptively secure unbounded* ABE with *multi-use* in the standard model.

References	KP/CP	Large univ.	Static assump.	Non-monoton.	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{sk}\rvert$
Att14 [6], Att16 [7], AC17a [3]	KP	✓			$O(1)$	$O(t)$	$O(n)$
KW19 [25]	KP		✓		$O(1)$	$O(t)$	$O(n)$
Att19 [8]	KP	✓		✓(OSW)	$O(1)$	$O(t)$	$O(n)$
Our 2	KP	✓	✓	✓(OSWOT)	$O(1)$	$O(t)$	$O(n)$
AY15 [11], Att16 [7], AC17a [3]	CP	✓			$O(1)$	$O(n)$	$O(t)$
Att19 [8]	CP	✓		✓(OSW)	$O(1)$	$O(n)$	$O(t)$
Our 2	CP	✓	✓	✓(OSWOT)	$O(1)$	$O(n)$	$O(t)$

Note: $t = \lvert$attribute set$\rvert$, n is the input length of a Boolean formula. $\lvert\mathsf{pk}\rvert$, $\lvert\mathsf{ct}\rvert$, $\lvert\mathsf{sk}\rvert$ is the size of public key, ciphertext, and secret key, respectively.

4. The first (adaptively secure) KP/CP-ABE with constant-size secret keys/ciphertexts under MDDH for monotone Boolean formulae, respectively.

Note that almost all previous ABE with constant-size ciphertexts or keys rely on q-type assumptions [1, 3, 6–9, 11], even when considering only selective security. There are only two exceptions: KP-ABE with constant-size ciphertexts of [16, 33], but these only achieves semi-adaptive security.

Discussion. We first clarify that our new ABE schemes are hard to obtain even if given the recent groundbreaking work by Kowalczyk and Wee (KW), who solved the multi-use problem in the adaptive setting and also presented an unbounded KP-ABE scheme with multi-use [25]. The crucial fact is that their scheme is a *small-universe* one. This is since their scheme relies on the bilinear expansion lemma [15] to obtain unboundedness, which causes security loss of attribute domain size. Tomida *et al.* recently presented completely unbounded ABE schemes, but they rely on random oracles to achieve unboundedness [34]. Thus, obtaining our completely unbounded ABE even for *monotone* Boolean formulae requires a new approach. Note that large-universe ABE has a significant advantage over small-universe one, even if the latter is unbounded. A clear advantage is that only large-universe ABE

Table 4. Comparison among ABE with *constant-size ciphertexts* ($|\mathsf{ct}| = O(1)$).

References	KP /CP	Large univ.	Adapt. security	Static assumptn.	Non-monoton.	Prime-order	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{sk}\rvert$
ALP11 [10]	KP	✓			✓(OSW)	✓	$O(T)$	$O(Tn)$
Att14 [6]	KP	✓	✓				$O(T)$	$O(Tn)$
CW14 [16]	KP			✓			$O(T)$	$O(Tn)$
Tak14 [33]	KP	✓		✓	✓(OSW)	✓	$O(T)$	$O(Tn)$
Att16 [7]	KP	✓	✓			✓	$O(T)$	$O(Tn)$
AC17a [3]	KP	✓	✓			✓	$O(T)$	$O(Tn)$
Att19 [8]	KP	✓	✓		✓(OSW)	✓	$O(T^2)$	$O(T^3n)$
Ours 3	KP	✓	✓	✓	✓(OSW)	✓	$O(T)$	$O(Tn)$
AHY15 [9]	CP	✓	✓		✓(OSW)	✓	$O((TN)^2\lambda)$	$O((TN)^4\lambda^2)$
AC16 [1]	CP			✓		✓	$O(N(T+M))$	$O(N^2T+NM)$
Att19 [8]	CP	✓	✓		✓(OSW)	✓	$O(N^2+NM)$	$O(t(N^3+N^2M))$
Ours 5	CP	✓	✓	✓		✓	$\tilde{O}((M+T\lambda)^2)$	$\tilde{O}((M+T\lambda)^4)$

Note: $t = |\text{attribute set}|$, n is the input length of a Boolean formula, while T, N are the maximum bound for t, n, respectively (if required). M is the maximum bound for the size of Boolean formulae (if required). λ is the security parameter, *i.e.,* $\lambda = \lceil \log p \rceil$. All the schemes shown in the table are large-universe ABE.

Table 5. Comparison among ABE with *constant-size keys* ($|\mathsf{sk}| = O(1)$).

References	KP /CP	Large univ.	Adapt. security	Static assumptn.	Non-monoton.	Prime-order	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ct}\rvert$
AY15 [11]	CP	✓	✓				$O(T)$	$O(Tn)$
Att16 [7]	CP	✓	✓			✓	$O(T)$	$O(Tn)$
AC17a [3]	CP	✓	✓			✓	$O(T)$	$O(Tn)$
Att19 [8]	CP	✓	✓		✓(OSW)	✓	$O(T^2)$	$O(T^3n)$
Ours 4	CP	✓	✓	✓	✓(OSW)	✓	$O(T)$	$O(Tn)$
AHY15 [9]	KP	✓	✓		✓(OSW)	✓	$O((TN)^2\lambda)$	$O((TN)^4\lambda^2)$
Att19 [8]	KP	✓	✓		✓(OSW)	✓	$O(N^2+NM)$	$O(t(N^3+N^2M))$
Ours 6	KP	✓	✓	✓		✓	$\tilde{O}((M+T\lambda)^2)$	$\tilde{O}((M+T\lambda)^4)$

Note: we refer to the same note as Table 4.

allows us to use any string as an attributes via collision-resistant hash functions, which has a super-poly size range.

We next highlight that our ABE for a new unified type of non-monotonicity has a practically meaningful functionality as well as a theoretical interest. Intuitively, it allows more desirable blacklisting access controls in dynamic systems where new attributes can be added on into the system *after deployment*. We will describe this with more details in §6. We remark that, in small universe ABE, we can use monotone ABE as non-monotone ABE by preparing both positive and negative attributes [30]. However, this is not the case in large-universe ABE since we cannot attach an exponentially large number of negative attributes to ciphertexts or secret keys. Hence, non-monotone ABE is essentially different from monotone ABE and generally more difficult to obtain in the context of large-universe ABE. Furthermore, the predicate for our new non-monotonicity is more expressive than previous ones, namely, OSW and OT type, and thus it seems extremely hard to construct such an ABE scheme from standard assumptions without a modular framework.

From the above observations, we believe that it is a quite challenging and important task to devise a framework that allows us to construct such ABE schemes in a modular manner from standard assumptions.

1.2 Technical Overview of Our Framework

We first describe the three main predicate transformations/compositions similarly to [8], namely, the Dual, the KP augmentation, and the Direct sum. Consider a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and a set $\mathcal{P} = \{\mathsf{P}_1, \dots, \mathsf{P}_k\}$ of predicates. We define a domain $\bar{\mathcal{Y}} := \bigcup_{i\in\mathbb{N}}(\mathcal{Y}^i \times \mathcal{F}_i)$, where $\mathcal{F}_i$ is a class of functions $\{0,1\}^i \to \{0,1\}$. We define domains $\hat{\mathcal{X}} = \bigcup_{i\in\mathbb{N}}\{i\} \times \mathcal{X}_i$ (and similarly for $\hat{\mathcal{Y}}$).

- The dual predicate $\mathsf{Dual}[\mathsf{P}] : \mathcal{Y} \times \mathcal{X} \to \{0,1\}$ is defined as $\mathsf{Dual}[\mathsf{P}](y,x) \; = \; \mathsf{P}(x,y)$.
- The KP augmentation predicate $\mathsf{KP1}[\mathsf{P}] : \mathcal{X} \times \bar{\mathcal{Y}} \to \{0,1\}$ is defined as follows. For $x \in \mathcal{X}$ and $Y = ((y_1,\dots,y_n), f) \in \bar{\mathcal{Y}}$, we set $\mathsf{KP1}[\mathsf{P}](x,Y) = f(\mathsf{P}(x,y_1),\dots,\mathsf{P}(x,y_n))$.
- The direct sum $\mathsf{DS}[\mathcal{P}] : \hat{\mathcal{X}} \times \hat{\mathcal{Y}} \to \{0,1\}$ is defined as: $\mathsf{DS}[\mathcal{P}]((i,x),(j,y)) = 1$ iff $i = j$ and $\mathsf{P}_i(x,y) = 1$.

We briefly note about a main composition as considered in [8], namely, the KP augmentation over *predicate sets*, $\mathsf{KP}[\mathcal{P}]$. For $X = \{(i_1,x_1),\dots,(i_t,x_t)\}$ and $Y = ((j_1,y_1),\dots,(j_n,x_n),f)$, we set $\mathsf{KP}[\mathcal{P}](X,Y) = f(b_1,\dots,b_n)$, where $b_k = 1$ iff there exists $(i,x) \in X$ such that $i = j_k$ and $\mathsf{P}_k(x,y_k) = 1$. We call it *static* if f is fixed (and hence so does n), otherwise it is *dynamic* over the class of f. We call it *unbounded* if n is unbounded. Attrapdung [8] showed that $\mathsf{Dual}, \mathsf{KP1}, \mathsf{DS}$ implies KP; therefore, we focus on the former three in this paper.

Background on PES. We now briefly recall PES [6], as refined in [3]. Informally, a PES for $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is represented by a variable α, five vectors of variables $(\mathbf{w},\mathbf{s},\hat{\mathbf{s}},\mathbf{r},\hat{\mathbf{r}})$, and two sets of polynomials (called ciphertext and key encodings, resp.) on these variables $(\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w}), \mathbf{k}_y(\alpha,\mathbf{r},\hat{\mathbf{r}},\mathbf{w}))$ that depend on $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, respectively. We require that $\mathbf{s}$ contains a variable s_0. Let $N = p_1p_2$ for primes p_1, p_2, and $e : G \times H \to G_\mathsf{T}$ be bilinear groups of order N. Let g_i, h_i be generators of the subgroups G_i, H_i of order p_i for $i \in \{1,2\}$, respectively, and $g = g_1g_2, h = h_1h_2$. Then, an ABE scheme in composite-order groups based on PES can be described as follows: $\mathsf{pk} = (g_1^{\mathbf{w}}, e(g_1,h)^\alpha)$ and

$$\mathsf{ct}_x = (g_1^{\mathbf{s}}, g_1^{\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w})}, e(g_1,h)^{s_0\alpha}m), \qquad\qquad \mathsf{sk}_y = (h_1^{\mathbf{r}}, h_1^{\mathbf{k}_y(\alpha,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})} h_2^{\mathbf{k}_y(\alpha,\mathbf{0},\hat{\mathbf{r}},\mathbf{0})}),$$

where $(\alpha,\mathbf{w},\mathbf{s},\hat{\mathbf{s}},\mathbf{r},\hat{\mathbf{r}}) \leftarrow \mathbb{Z}_N^t$ (t is the total number of the variables). We require that each polynomial of $\mathbf{c}_x$ is a linear combination of monomials s_iw_j and $\hat{s}_k$ (where $s_i \in \mathbf{s}$, $\hat{s}_k \in \hat{\mathbf{s}}$, $w_j \in \mathbf{w}$). This yields the linearity of $\mathbf{c}_x$ over $\mathbf{s}, \hat{\mathbf{s}}$, when fixing $\mathbf{w}$. Analogous properties go for key encodings. As an example, a PES for IBE [6] has the form $\mathbf{c}_x = s_0(w_1x + w_2)$, $\mathbf{k}_y = \alpha + r_1(w_1y + w_2)$, where $\mathbf{w} = (w_1, w_2)$, $\mathbf{s} = s_0$, $\mathbf{r} = r_1$ (and no $\hat{\mathbf{s}}, \hat{\mathbf{r}}$). In what follows in this section, we write $\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w})$ and $\mathbf{k}_y(\alpha,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})$ to implicitly include $\mathbf{s}$ and $\mathbf{r}$, respectively.

Our Goal: Three Main Implications. Since the symbolic property is useless without the q-ratio assumption, we need a completely different new notion on PES that is preserved via the transformations, and that, at the same time, implies the adaptive security of the induced ABE scheme under standard assumptions. To this end, in this work, we introduce a new central notion called Key-Encoding Indistinguishability for PES, denoted $\mathsf{KE\text{-}ind}$. Our goal is to design $\mathsf{KE\text{-}ind}$ in such a way that the following theorems (stated informally below) hold. The first states the preservation of $\mathsf{KE\text{-}ind}$ under the transformation. The second states that $\mathsf{KE\text{-}ind}$ implies adaptively secure ABE under MDDH.

Informal Theorem 1. *For a composition* $\mathsf{C} \in \{\mathsf{Dual}, \mathsf{DS}, \mathsf{KP1}\}$*, if there exists a PES for* P *that satisfies* $\mathsf{KE\text{-}ind}$*, then there exists a PES for* $\mathsf{C}[\mathsf{P}]$ *that satisfies* $\mathsf{KE\text{-}ind}$ *under MDDH. (Note that for* DS*, its input is a predicate set* $\mathcal{P}$*.)*

Informal Theorem 2. *If there exists a PES for* P *that satisfies* $\mathsf{KE\text{-}ind}$*, then there exists an adaptively secure ABE scheme for* P *under MDDH.*

The third theorem finally tells us how to achieve $\mathsf{KE\text{-}ind}$ via the existing information-theoretic notion of PES called perfect master-key hiding (PMH) of PES as defined in [6]. PMH requires that the following two distributions are identical with respect to $(\alpha,\mathbf{w},\mathbf{s},\hat{\mathbf{s}},\mathbf{r},\hat{\mathbf{r}}) \leftarrow \mathbb{Z}_N^t$:

$$\{\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w}), \mathbf{k}_y(\alpha,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})\} \;\text{ and }\; \{\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w}), \mathbf{k}_y(0,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})\}. \tag{1}$$

Informal Theorem 3. *If a PES satisfies the PMH property, then the same PES also satisfies* $\mathsf{KE\text{-}ind}$ *under MDDH.*

From these theorems, we have the following corollary.

Informal Corollary 1. *If there exists a PES for* P *satisfying the PMH property, then there exists an adaptively secure ABE for the composed predicate* $\mathsf{C}[\mathsf{P}]$ *under MDDH, where* $\mathsf{C} \in \{\mathsf{Dual}, \mathsf{DS}, \mathsf{KP1}, \mathsf{KP}\}$. *(For* DS, KP*, we consider a set* $\mathcal{P}$*.)*

We can thus start from such information-theoretic PESs for basic predicates in *e.g.,* [5,6], such as IBE, and obtain adaptively secure ABE for composed predicates. Moreover, due to the preservation of $\mathsf{KE\text{-}ind}$, we can apply conversions *recursively* to obtain complex predicates as desired.

To obtain these theorems, it remains to properly design $\mathsf{KE\text{-}ind}$.

Designing Key-Encoding Indistinguishability. For simplicity, we explain our framework in composite-order bilinear groups in this overview since we can basically convert ABE constructions in composite-order groups into those in prime-order groups via the framework by Chen *et al.* [14,15,20]. Note that the MDDH assumption in prime-order groups corresponds to the subgroup (SG) assumptions in composite-order groups (see e.g., [15]).

Our starting point is to define $\mathsf{KE\text{-}ind}$ to be exactly the *computationally master-key hiding* (CMH) property [6], which is a relaxed notion of PMH (and we would obtain Theorem 3 above). We say that a PES Γ specified by $(\alpha, \mathbf{w}, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{c}_x, \mathbf{k}_y)$ for P satisfies CMH if the following advantage of $\mathcal{A}$ is negligible:

$$\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Gamma}(\lambda) = \left| \mathsf{Pr}\left[\beta = \beta' \,\middle|\, \begin{array}{l} \beta \leftarrow \{0,1\} \\ \beta' \leftarrow \mathcal{A}^{\mathsf{cO}(\cdot), \mathsf{kO}_\beta(\cdot)}(g_1, g_2, h_1, h_2) \end{array} \right] - \frac{1}{2} \right|,$$

where the ciphertext encoding oracle cO takes $x \in \mathcal{X}$ and outputs $g_2^{\mathbf{c}_x(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})}$, while the key encoding oracle kO_β takes $y \in \mathcal{Y}$ and outputs $h_2^{\mathbf{k}_y(\beta\alpha, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})}$, where $\alpha, \mathbf{w}, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{r}, \hat{\mathbf{r}}$ are random. Here $\mathcal{A}$ can query each oracle once with $R(x, y) = 0$. Attrapadung showed that if we have a PES for P with CMH, then we can obtain an adaptively secure ABE scheme for P assuming the SG assumption [6] (this implies Theorem 2). Thus, if we could show that CMH is preserved via the transformations (this would imply Theorem 1), we would achieve the goal.

Unfortunately, we quickly found out that this approach fails; in particular, we do not know how to preserve CMH via the $\mathsf{KP1}$ transformation. Assume that we use the same $\mathsf{KP1}$ transformation as in [8], which transforms a PES Γ for P to a PES Γ' for $\mathsf{KP1}[\mathsf{P}]$ to be exactly the same as Γ except that

$$\mathbf{k}'_Y(\alpha, \mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}) = \{\mathbf{k}_{y_i}(\sigma_i, \mathbf{r}_i, \hat{\mathbf{r}}_i, \mathbf{w})\}_{i \in [n]}$$

and $\mathbf{r}' = \{\mathbf{r}_i\}_{i \in [n]}$, $\hat{\mathbf{r}}' = \{\hat{\mathbf{r}}_i\}_{i \in [n]}$, where $\{\sigma_i\}_{i \in [n]}$ are secret shares of α with respect to f. (Here, primed variables are for Γ'.) Our goal here is to construct a reduction that breaks CMH of Γ internally using an adversary that breaks CMH of Γ'. One hopeful strategy is to limit f to Boolean formulae and consider a series of hybrids as the KW framework [25]. However, this idea does not work as the reduction cannot simulate $\{h_2^{\mathbf{k}_{y_i}(\sigma_i, \mathbf{r}_i, \hat{\mathbf{r}}_i, \mathbf{w})}\}_{i \neq j}$ when randomizing $h_2^{\mathbf{k}_{y_j}(\sigma_j, \mathbf{r}_j, \hat{\mathbf{r}}_j, \mathbf{w})}$ due to the absence of $h_2^{\mathbf{w}}$. Including $h_2^{\mathbf{w}}$ in the input of the CMH adversary does not solve the problem since this makes PMH not imply CMH, and Theorem 3 does not hold in such a definition (observe that in Eq. (1), $\mathbf{w}$ is not given out). Our next observation here is that we will need a property on indistinguishability of H_2 elements where the output of kO_β is simulatable *without* $h_2^{\mathbf{w}}$.

First Step. Our first idea is to make the outputs of cO and kO_β use *entire* groups G, H instead of only *subgroups* G_2, H_2. A new candidate property (say, $\mathsf{Cand1}$) for Γ is then defined as follows:

$$\mathsf{Adv}^{\mathsf{Cand1}}_{\mathcal{A},\Gamma}(\lambda) = \left| \mathsf{Pr}\left[\beta = \beta' \,\middle|\, \begin{array}{l} \beta \leftarrow \{0,1\}, \ \ \mathbf{w} \leftarrow \mathbb{Z}_N^\omega \\ \beta' \leftarrow \mathcal{A}^{\mathsf{cO}(\cdot), \mathsf{kO}_\beta(\cdot)}(g_1, h_1, h_2, g_1^{\mathbf{w}}, h_1^{\mathbf{w}}) \end{array} \right] - \frac{1}{2} \right|,$$

where $g^{\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w})} \leftarrow \mathsf{cO}(x)$ and $h_1^{\mathbf{k}_y(0,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})} h_2^{\mathbf{k}_y(\beta\alpha,\mathbf{0},\hat{\mathbf{r}},\mathbf{0})} \leftarrow \mathsf{kO}_\beta(y)$ where $\alpha, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{r}, \hat{\mathbf{r}}$ are random. Crucially, now, g_2 is not given out to $\mathcal{A}$.

$\mathsf{Cand1}$ implies an adaptive security of the ABE scheme from Γ (and we obtain Theorem 2). Intuitively, the indistinguishability of the H_2 elements in the output of kO_β implies the indistinguishability between normal and semi-functional keys, which then implies the adaptive security of the ABE scheme via the dual system technique [35]. Next, $\mathsf{Cand1}$ can be shown to be implied by PMH and the SG assumption (and we obtain Theorem 3) as follows (also recall linearity of $\mathbf{k}_y$):

$$h_1^{\mathbf{k}_y(0,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})} h_2^{\mathbf{k}_y(0,\mathbf{0},\hat{\mathbf{r}},\mathbf{0})} \underset{\text{SG}}{\approx_c} - \cdot h_2^{\mathbf{k}_y(0,\underline{\mathbf{r}},\hat{\mathbf{r}},\underline{\mathbf{w}})} \underset{\text{PMH}}{\approx_s} - \cdot h_2^{\mathbf{k}_y(\underline{\alpha},\mathbf{r},\hat{\mathbf{r}},\mathbf{w})} \underset{\text{SG}}{\approx_c} - \cdot h_2^{\mathbf{k}_y(\alpha,\underline{\mathbf{0}},\hat{\mathbf{r}},\underline{\mathbf{0}})}.$$

Note that "$-$" is the same element in H_1, and $\approx_c, \approx_s$ are computational and statistical indistinguishability, respectively. The purpose for making g_2 absent in $\mathcal{A}$'s input is to use the SG assumption that claims $h_1^{\mathbf{r}} \approx_c h^{\mathbf{r}}$. In this way, we can prove that $\mathsf{Cand1}$ is preserved in $\mathsf{KP1}$ for Boolean formulae by extending the KW framework. Intuitively, the reduction goes through as it can simulate $K_i = h_1^{\mathbf{k}_{y_i}(0,\mathbf{r}_i,\hat{\mathbf{r}}_i,\mathbf{w})} h_2^{\mathbf{k}_{y_i}(\sigma_i,\mathbf{0},\hat{\mathbf{r}}_i,\mathbf{0})}$ without $h_2^{\mathbf{w}}$ (observe that there is no $\mathbf{w}$ in the exponent to h_2 in K_i).

However, it turns out that $\mathsf{Cand1}$ is not preserved in Dual. Assume that we use the same Dual transformation as in [3], which transforms a PES Γ for P to a PES $\overline{\Gamma}$ for $\mathsf{Dual}[\mathsf{P}]$ as follows: first let the variables for $\overline{\Gamma}$ be $\mathbf{w}' = (w_0, \mathbf{w})$, $\mathbf{s}' = (s_{\mathsf{new}}, \mathbf{r})$, $\hat{\mathbf{s}}' = \hat{\mathbf{r}}$, $\mathbf{r}' = \mathbf{s}$, $\hat{\mathbf{r}}' = \hat{\mathbf{s}}$ and define the two encodings for $\overline{\Gamma}$ as

$$\mathbf{c}'_y(\mathbf{s}', \hat{\mathbf{s}}', \mathbf{w}') = \mathbf{k}_y(s_{\mathsf{new}} w_0, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{w}), \qquad \mathbf{k}'_x(\alpha, \mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}') = (\mathbf{c}_x(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}), \alpha - s_0 w_0),$$

where w_0, s_{new} are new variables, and s_{new} takes a role of s_0 in $\overline{\Gamma}$. To prove the preservation of $\mathsf{Cand1}$ in Dual, we need to construct a reduction $\mathcal{R}$ that breaks $\mathsf{Cand1}$ of Γ internally using an adversary $\mathcal{A}$ against ($\mathsf{Cand1}$ of) $\overline{\Gamma}$. A crucial fact here is that the roles of G and H are "switched", that is, $\mathcal{R}$ uses its input G and H as H and G for the input of $\mathcal{A}$, respectively. This is since $\mathcal{R}$ needs the reply of $\mathsf{cO}^{\mathcal{R}}$ to answer $\mathcal{A}$'s query to $\mathsf{kO}^{\mathcal{A}}$ (and analogously for $\mathsf{kO}^{\mathcal{R}}$ to $\mathsf{cO}^{\mathcal{A}}$). Now the problem arises as $\mathcal{R}$ does not possess g_2, but this very term will be needed to supply to $\mathcal{A}$'s input as h_2 (recall the "switching" of G and H). Also recall that h_2 was necessary to prove Theorem 2 (to simulate semi-functional keys).

Second Step. To solve the above problem, instead of preserving the *same* property from Γ to $\overline{\Gamma}$, we will establish an implication over *slightly different* properties on Γ and $\overline{\Gamma}$. Namely, we use more subgroups by letting $N = p_1 \cdots p_z$ and parametrize the candidate property as (z,ℓ)-$\mathsf{Cand2}$, where $z, \ell \in \mathbb{N}$ s.t. $z \geq \ell$. Defining bilinear groups $e : G \times H \to G_{\mathsf{T}}$ of order N and its subgroups naturally, we then define $\mathsf{Adv}^{(z,\ell)\text{-}\mathsf{Cand2}}_{\mathcal{A},\Gamma}(\lambda)$ as

$$\left| \Pr\left[\beta{=}\beta' \,\middle|\, \begin{array}{l} \beta \leftarrow \{0,1\}, \quad \mathbf{w} \leftarrow \mathbb{Z}_N^{\omega} \\ \beta' {\leftarrow} \mathcal{A}^{\mathsf{cO}(\cdot),\mathsf{kO}_\beta(\cdot)}(g_1, h_1, g_{\ell+1}, \ldots, g_z, h_\ell, \ldots, h_z, g_1^{\mathbf{w}}, h_1^{\mathbf{w}}) \end{array} \right] - \frac{1}{2} \right| \tag{2}$$

where $g^{\mathbf{c}_x(\mathbf{s},\hat{\mathbf{s}},\mathbf{w})} \leftarrow \mathsf{cO}(x)$ and $h_1^{\mathbf{k}_y(0,\mathbf{r},\hat{\mathbf{r}},\mathbf{w})} h_\ell^{\mathbf{k}_y(\beta\alpha,\mathbf{0},\hat{\mathbf{r}},\mathbf{0})} \leftarrow \mathsf{kO}_\beta(y)$. In this way, we have that g_ℓ is absent (generalizing the absence of g_2, so as to establish Theorem 3 as in the first step), but now, at the same time, we can also potentially establish the implication over Dual that $(z, \ell-1)$-$\mathsf{Cand2}$ of Γ implies (z,ℓ)-$\mathsf{Cand2}$ of $\overline{\Gamma}$ for $\ell \geq 2$ in the sense that the reduction $\mathcal{R}$ possesses $g_\ell, \ldots, g_z$ (as per the former notion) which can be used to exactly simulate $h_\ell, \ldots, h_z$ (giving to the adversary $\mathcal{A}$ against the latter notion), where we recall the switching of G and H.

Final Step. In the above, we generalize the functionality of the subgroups G_2, H_2 directly to G_ℓ, H_ℓ and hence obtain the above design of the oracle kO. However, this design fails when we try to use the reply of $\mathsf{cO}^{\mathcal{R}}$ to answer $\mathcal{A}$'s query to $\mathsf{kO}^{\mathcal{A}}$ (as presumably required in the reduction). This is since the former is an element of the entire group, while the latter is in the subgroup with generators h_1, h_ℓ; however, $\mathcal{A}$ possesses $g_{\ell+1}$ and thus can simply distinguish the two. A similar failure occurs analogously when relating $\mathsf{kO}^{\mathcal{R}}$ to $\mathsf{cO}^{\mathcal{A}}$. To solve this, we need to re-design also the two oracles carefully (satisfying not only this particular preservation of Dual that we are discussing but also all the required 3 theorems). To this end, our solution is to define

$$\begin{aligned} g_1^{\mathbf{c}_x(\mathbf{s},\mathbf{0},\mathbf{w})} g_{[2,\ell]}^{\mathbf{c}_x((s_0,\mathbf{0}),\mathbf{0},\mathbf{w})} g^{\mathbf{c}_x(\mathbf{0},\hat{\mathbf{s}},\mathbf{0})} &\quad \leftarrow \mathsf{cO}(x), \\ h_1^{\mathbf{k}_y(0,\mathbf{r},\mathbf{0},\mathbf{w})} h_\ell^{\mathbf{k}_y(\beta\alpha,\mathbf{0},\mathbf{0},\mathbf{0})} h^{\mathbf{k}_y(0,\mathbf{0},\hat{\mathbf{r}},\mathbf{0})} &\quad \leftarrow \mathsf{kO}_\beta(y), \end{aligned}$$

and also additionally give out $T = (g_{[1,\ell]}, \ldots, g_{[1,z]}, h_{[1,\ell+1]}, \ldots, h_{[1,z]})$ (as inputs to $\mathcal{A}$ in Eq. (2)), where we denote $g_{[a,b]} = g_a \cdots g_b$ for $a \le b$. Intuitively, the forms of $\mathsf{cO}^{\mathcal{R}}$ and $\mathsf{kO}^{\mathcal{A}}$ are now similar, except the difference lying in the subgroups with indexes $2, \ldots, \ell - 1$, and we observe that the adversary does not possess an element from these subgroups so as to distinguish the two; therefore, we can use the former to simulate the latter, under the SG assumption. The additional input T is essential for the other oracle simulation (from $\mathsf{kO}^{\mathcal{R}}$ to $\mathsf{cO}^{\mathcal{A}}$). Crucially, giving out individual generators such as $g_2, \ldots, g_\ell$ would destroy the "absence" requirement (essential for Theorem 3); while, on the other hand, giving out the elements like $g_{[1,i]}$ do work.

This completes our design rational of (z,ℓ)-KE-ind (in the composite-order-groups flavor). Note that ℓ is incremented by 1 after applying one Dual conversion. Starting from $(z,1)$-KE-ind, we have that $z-1$ is the maximum number of Dual applications. Thus, by choosing z depending on the number of dual applications to obtain a target predicate P, we can instantiate a secure ABE scheme for P. Also note that (z,ℓ)-KE-ind will require $\mathbf{s}$ to consist of only s_0 so that it is implied by PMH. We call it single-variable PMH. Note that PESs with single-variable PMH are still more general encodings than predicate encodings [5, 37].

1.3 Technical Comparison to Previous Unbounded ABE

Our framework allows us to modularly construct unbounded ABE schemes. Thus, one may wonder how our framework compares to previous unbounded ABE schemes from static assumptions [15, 25, 27, 29]. Basically, these ABE schemes rely on so-called "nested dual system technique", in which entropy in secret keys is increased via entropy propagation between a secret key and ciphertext. All these works uses the IBE predicate as a source of entropy.

Intuitively, our framework can be viewed as decomposing the entropy propagation into modular parts, namely, the PMH of a PES for IBE, the KP1 transform, and the Dual transform (recall that our framework applies KP1 and Dual to the IBE predicate in a nested manner to achieve our completely unbounded monotone scheme, similarly to [8]). This predicate transformations implicitly trace a similar hybrid sequence to that by Lewko and Waters (LW) [27], borrowing the power of the KW framework (the piecewise guessing framework) to do it in the adaptive setting. An important fact here is that our framework uses the KW framework in a nested manner. Intuitively, this is the reason why our ABE schemes can be constructed as large-universe constructions similarly to the LW unbounded scheme. On the other hand, the KW unbounded scheme [25] is obtained by directly applying the KW framework (not in a nested manner) to the ABE scheme of [15], which is based on the bilinear expansion lemma [15]. Consequently, their scheme is limited to small-universe since the bilinear expansion lemma needs "pre-processing" of *all* attributes in the domain. In fact, the universe size U is reflected as a linear term in the security loss (and hence U cannot be super-polynomially large) for the KW scheme (see Table 6).

Another advantage of our framework over the KW scheme is that we do not use the subgroup DDH assumption [15], which requires a k-dimensional semi-functional space for the k-Lin assumption. In contrast, 1-dimensional semi-functional spaces suffice for our framework. This yields asymptotically smaller ciphertexts and keys than the KW scheme (asymptotic in k, see Table 6).

Finally, unbounded ABE by Okamoto and Takashima has a large universe [29]. Thus, if we can apply the KW framework to their scheme, we would obtain a completely unbounded scheme, but it seems not so straightforward.

Table 6. Comparison with unbounded KP-ABE from $\mathcal{D}_k$-MDDH by KW19 [25].

References	Security loss	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{sk}\rvert$
KW19 [25]	$O(Uq_{\mathsf{sk}})2^{O(B)}$	$(5k^2+k)\lvert G_1\rvert + k\lvert G_{\mathsf{T}}\rvert$	$((3k+1)t+2k+1)\lvert G_1\rvert + \lvert G_{\mathsf{T}}\rvert$	$((5k+2)n+(2k+1)m)\lvert G_2\rvert$
Ours 1 (§A.1)	$O(q_{\mathsf{sk}})2^{O(B)}$	$(4k^2+8k)\lvert G_1\rvert + k\lvert G_{\mathsf{T}}\rvert$	$((2k+4)t+k+2)\lvert G_1\rvert + \lvert G_{\mathsf{T}}\rvert$	$(3k+6)n\lvert G_2\rvert$

Note: U is the attribute domain size, q_{sk} is the maximum number of secret key queries, B is the maximum depth of formulae, $t = \lvert$attribute set$\rvert$, m and n are the number of gates and the input length of a formula, respectively.

2 Preliminaries

Notation. For a natural number $m, n \in \mathbb{N}$, $[m]$ denotes a set $\{1, \dots, m\}$, $[m]^+$ denotes a set $\{0, \dots, m\}$, and $[m, n]$ denotes a set $\{m, \dots, n\}$. For a set S, $s \leftarrow S$ denotes that s is uniformly chosen from S. We treat vectors as column vectors unless specified otherwise. For a generator g_i of a cyclic group G_i of order p and $a \in \mathbb{Z}_p$, $[a]_i$ denotes g_i^a. Furthermore, for a matrix $\mathbf{A} = (a_{j,\ell})_{j,\ell}$ over $\mathbb{Z}_p$, $[\mathbf{A}]_i$ denotes a matrix over G_i whose (j, ℓ)-th entry is $g_i^{a_{j,\ell}}$. For vectors $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{Z}_p^n$, let $e([\mathbf{x}]_1, [\mathbf{y}]_2) = e(g_1, g_2)^{\langle \mathbf{x}, \mathbf{y} \rangle}$ be a function that computes the inner product on the exponent by $\prod_{i\in[n]} e([x_i]_1, [y_i]_2)$. A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and denotes $f(\lambda) \le \mathsf{negl}(\lambda)$. For families of distributions $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$, we denote $X \approx_c Y$ (resp. $X \approx_s Y$) as computational indistinguishability (resp. statistical indistinguishability). For an interactive game G, $\langle \mathcal{A}, \mathsf{G} \rangle$ denotes the output of $\mathcal{A}$ in G.

Matrix notation. Throughout the paper, we use the following matrix notation. For a regular matrix $\overline{\mathbf{M}} \in \mathsf{GL}_{k+\zeta}(\mathbb{Z}_p)$, we define $\mathbf{M}$, $\mathbf{m}_i$, $\mathbf{M}^*$, and $\mathbf{m}_i^*$ as follows. $\mathbf{M}$ and $\mathbf{m}_i$ denote a matrix and a vector consist of the first k columns and the $(k+i)$-th column of $\overline{\mathbf{M}}$, respectively. Similarly, $\mathbf{M}^*$ and $\mathbf{m}_i^*$ denote a matrix and vector consist of the first k columns and the $(k+i)$-th column of $(\overline{\mathbf{M}}^\top)^{-1}$, respectively. We have the relations, $\mathbf{M}^\top \mathbf{m}_i^* = \mathbf{0}$ and $\mathbf{m}_i^\top \mathbf{m}_i^* = 1$ for $i \in [\zeta]$. We also uses the following notations:

$$\mathsf{span}(\mathbf{M}, \mathbf{m}_1, \dots, \mathbf{m}_n) = \{\mathbf{v} \mid \exists \mathbf{u} \in \mathbb{Z}_p^{k+n}, \mathbf{v} = (\mathbf{M}||\mathbf{m}_1|| \dots ||\mathbf{m}_n)\mathbf{u}\},$$
$$\mathsf{Ker}(\mathbf{M}, \mathbf{m}_1, \dots, \mathbf{m}_n) = \{\mathbf{v} \mid (\mathbf{M}||\mathbf{m}_1|| \dots ||\mathbf{m}_n)^\top \mathbf{v} = \mathbf{0}\}.$$

2.1 Basic Definitions and Tools

Boolean Formula and NC^1. A monotone Boolean formula can be represented by a Boolean circuit of which all gates have fan-in 2 and fan-out 1. More precisely, we specify a monotone Boolean formula by a tuple $f = (n, w, m, G)$ where $n, w, m \in \mathbb{N}$ represents the number of input wires, the number of all wires (including the input wires), and the number of gates, respectively, while $G : [m] \to \{\text{AND, OR}\} \times [w]^3$ is a function that specifies the gate type, the two incoming wires, and the outgoing wire of each gate. To specify G, we first let all the wires and gates to be numbered. The wire numbers range from 1 to w; while those of gates range from 1 to m. For each gate $i \in [m]$, the information $G(i) = (T, a, b, c)$ tells us that T is the type of the gate i, while a and b specify its incoming wires, and c specifies its outgoing wire. By convention, we always number the wires so that $a < b < c$. The computation of Boolean formula f on an input in $\{0, 1\}^n$ is defined naturally; we often abuse the notation and treat f as a function $f : \{0, 1\}^n \to \{0, 1\}$.

A non-monotone Boolean formula additionally contains NOT gates, which have fan-in 1 and fan-out 1. It is well-known that, via De Morgan's law, we can express any non-monotone Boolean formula by one in which all the NOT gates are placed on the input wires (and the number of gates of the latter formula is two times of that of the former). Hence, we can specify a non-monotone Boolean formula as a tuple $f = (n, w, m, G, \Sigma)$, where $\Sigma : [n] \to \{\text{Positive}, \text{Negative}\}$ naturally specifies if the input wire $i \in [n]$ is a negative one or not.

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.,* [25]). Thus, the circuit complexity class captured by Boolean formulae is equivalent to the class captured by log-depth Boolean formulae.

Definition 1 (Linear Secret Sharing Scheme). A linear secret sharing scheme (LSSS) for a function class $\mathcal{F}$ consists of two algorithms Share and Rec.

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

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

In particular, Rec computes a linear function on shares to reconstruct a secret; $\mathbf{h} = \sum_{x_i=1} a_i \mathbf{h}_i$ 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 $f(x) = 1$,

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

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

Definition 2 (Bilinear Groups). A description of bilinear groups $\mathbb{G}{=}(p, G_1, G_2, G_{\mathsf{T}}, g_1, g_2, e)$ 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 $e : G_1 \times G_2 \to G_{\mathsf{T}}$, which has two properties.

– (Bilinearity): $\forall h_1 \in G_1, h_2 \in G_2, a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$.
– (Non-degeneracy): For generators g_1, g_2; $g_{\mathsf{T}} = e(g_1, g_2)$ 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 a $\Omega(\lambda)$-bit prime p.

Definition 3 ($\mathcal{D}_{j,k}$-MDDH Assumption [19]). For $j > k$, let $\mathcal{D}_{j,k}$ be a matrix distribution over matrices in $\mathbb{Z}_p^{j \times k}$, which outputs a full-rank matrix with overwhelming probability. We can assume that, wlog, the first k rows of a matrix chosen from $\mathcal{D}_{j,k}$ form an invertible matrix. We consider the following distribution: $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{j,k}$, $\mathbf{v} \leftarrow \mathbb{Z}_p^k$, $\mathbf{t}_0 = \mathbf{A}\mathbf{v}$, $\mathbf{t}_1 \leftarrow \mathbb{Z}_p^j, P_{i,\beta} = (\mathbb{G}, [\mathbf{A}]_i, [\mathbf{t}_\beta]_i)$. We say that the $\mathcal{D}_{j,k}$-MDDH assumption holds with respect to $\mathcal{G}_{\mathsf{BG}}$ if, for any PPT adversary $\mathcal{A}$ and $i \in \{1,2\}$,

$$\mathsf{Adv}_{\mathcal{A},i}^{\mathcal{D}_{j,k}\text{-}\mathsf{MDDH}}(\lambda) = |\Pr[1 \leftarrow \mathcal{A}(P_{i,0})] - \Pr[1 \leftarrow \mathcal{A}(P_{i,1})]| \leq \mathsf{negl}(\lambda).$$

In what follows, we denote $\mathcal{D}_{k+1,k}$ by $\mathcal{D}_k$.

Uniform distribution Let $\mathcal{U}_{j,k}$ be a uniform distribution over $\mathbb{Z}_p^{j \times k}$. Then, the following hold with tight reductions: $\mathcal{D}_k$-MDDH $\Rightarrow$ $\mathcal{U}_k$-MDDH $\Rightarrow$ $\mathcal{U}_{j,k}$-MDDH.

Random self-reducibility We can obtain arbitrarily many instances of the $\mathcal{D}_k$-MDDH problem without additional security loss. For any $n \in \mathbb{N}$, we define the following distribution: $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{j,k}$, $\mathbf{v} \leftarrow \mathbb{Z}_p^k$, $\mathbf{V} \leftarrow \mathbb{Z}_p^{k \times n}$, $\mathbf{T}_0 = \mathbf{A}\mathbf{V}$, $\mathbf{T}_1 \leftarrow \mathbb{Z}_p^{j \times n}, P_{i,\beta} = (\mathbb{G}, [\mathbf{A}]_i, [\mathbf{T}_\beta]_i)$. The n-fold $\mathcal{D}_k$-MDDH assumption is similarly defined to the $\mathcal{D}_k$-MDDH assumption. Then, the n-fold $\mathcal{D}_k$-MDDH assumption is tightly reduced to the $\mathcal{D}_k$-MDDH assumption. That is, $\mathcal{D}_k$-MDDH $\Rightarrow$ n-$\mathcal{D}_k$-MDDH.

2.2 Attribute-Based Encryption

Predicate Family. Let $\mathsf{P} = \{\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\} \mid \kappa \in \mathcal{K}\}$ be a predicate family where $\mathcal{X}_\kappa$ and $\mathcal{Y}_\kappa$ denote "ciphertext attribute" and "key attribute" spaces. The index κ denotes a list of some parameters such as bounds on some quantities (hence $\mathcal{K}$ depends on that predicate). We often omit κ if the context is clear.

Definition 4 (Attribute-Based Encryption). An attribute-based encryption (ABE) scheme for a predicate family P consists of four algorithms:

$\mathsf{Setup}(1^\lambda, \kappa)$**:** It takes a security parameter 1^λ, and an index κ as inputs, and outputs a public key pk and a master secret key msk.

$\mathsf{Enc}(\mathsf{pk}, x, M)$**:** It takes pk, an attribute $x \in \mathcal{X}$ and a message $M \in \mathcal{M}$ as inputs, and outputs a ciphertext ct_x. (Note that we let $\mathcal{M}$ be specified in pk.)

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$**:** It takes $\mathsf{pk}, \mathsf{msk}$, and an attribute $y \in \mathcal{Y}$ as inputs, 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 as inputs, and outputs a message M' or a symbol $\perp$.

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

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

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 $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}}(\lambda) = \left| \mathsf{Pr}\left[\beta = \beta' \,\middle|\, \begin{array}{l} \beta \leftarrow \{0,1\} \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \kappa) \\ (x^*, M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot)}(\mathsf{pk}) \\ \mathsf{ct}_{x^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, x^*, M_\beta) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \cdot)}(\mathsf{ct}_{x^*}) \end{array} \right] - \frac{1}{2} \right|,$$

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

2.3 Piecewise Guessing Framework

We briefly recall the piecewise guessing framework by Kowalczyk and Wee [25], which is based on the framework by Jafargholi et al. [23]. The framework helps us to prove adaptive security of cryptographic schemes that are selectively secure.

Definition 5 (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 $h_0, \ldots, h_L : \{0,1\}^R \rightarrow \{0,1\}^{R'}$ 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. Now we are ready to state the adaptive security lemma.

Lemma 1 (Adaptive Security Lemma [25]). *Let G_0 and G_1 be adaptive interactive games and $\{\widehat{\mathsf{H}}^{h_i}\}_{0\le i\le L}$ 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 adversaries $\mathcal{A}$ and all $\iota \in L$, we have*

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

Then, we have

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

2.4 Pebbling Strategy for Boolean Formulae

A pebbling strategy for Boolean Formula is a guide of how to construct a series of hybrids in the piecewise guessing framework to prove a sort of adaptive security on a computational secret sharing scheme for Boolean Formulae.

Definition 6 (Pebbling Game). A player of the pebbling game is given a monotone Boolean formula $f : \{0,1\}^n \to \{0,1\}$ and input $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ such that $f(b) = 0$. The goal of the game is to reach the state where a pebble is placed on only the output gate, starting from the state with no pebbles on the Boolean formula f, following a pebbling rule. The rule is defined as follows.

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

Definition 7 (Pebbling Record). A pebbling record $\mathcal{R} = (r_0, \ldots, r_L) \in (\{0,1\}^{R'})^L$ is a list of all pebbling configuration that a player took from the start to the goal in the game. The 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. Furthermore, all pebbling configurations that the player took can be specified by an R'-bit string.

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

Lemma 2 (Pebbling Lemma [25]). *Let $f : \{0,1\}^n \to \{0,1\}$ be any monotone Boolean formula with a depth $d \le B$, 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.*

2.5 Embedding Lemma

For arguing implications among PESs, we use the embedding lemma. Such a lemma is already known and applied for arguing implications among ABE schemes [9, 13] and PES [8]. Here we capture that the embedding also preserves our new security notion for PES, namely, (ζ,ℓ)-$\mathsf{KE\text{-}ind}$, as well, in the lemma below.

Definition 8 ([8]). Let $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$, and $P'_{\kappa'} : \mathcal{X}'_{\kappa'} \times \mathcal{Y}'_{\kappa'} \to \{0,1\}$ be two predicate families, indexed by $\kappa = (N, \mathsf{par}) \in \mathcal{K}$ and $\kappa' = (N, \mathsf{par}') \in \mathcal{K}'$, respectively. We say that P' *can be embedded into* P if there exists three efficient mappings $f_\mathsf{p}, f_\mathsf{e}, f_\mathsf{k}$ where $f_\mathsf{p} : \mathcal{K}' \to \mathcal{K}$ maps $\kappa' = (N, \mathsf{par}') \mapsto \kappa = (N, \mathsf{par})$ and $f_\mathsf{e} : \mathcal{X}'_{\kappa'} \to \mathcal{X}_\kappa$, $f_\mathsf{k} : \mathcal{Y}'_{\kappa'} \to \mathcal{Y}_\kappa$ such that for all $x' \in \mathcal{X}'_{\kappa'}, y' \in \mathcal{Y}'_{\kappa'}$, we have:

$$P'_{\kappa'}(x', y') = 1 \quad \Longleftrightarrow \quad P_\kappa(f_\mathsf{e}(x'), f_\mathsf{k}(y')) = 1. \tag{3}$$

Lemma 3. *If P' can be embedded into P, then any PES for P secure in the sense of (ζ,ℓ)-$\mathsf{KE\text{-}ind}$ implies a PES for P' secure in the same sense.*

Proof sketch. Let Γ be a PES for P. We construct a PES Γ' for P' by simply defining $\mathsf{Param}'(\mathsf{par}') = \mathsf{Param}(f_\mathsf{p}(\mathsf{par}'))$, $\mathsf{EncCt}'(y', N) = \mathsf{EncCt}(f_\mathsf{e}(y'), N)$, and $\mathsf{EncKey}'(x', N) = \mathsf{EncCt}(f_\mathsf{k}(x'), N)$. Also define $\mathsf{Pair}'(x', y', N) = \mathsf{Pair}(f_\mathsf{k}(x'), f_\mathsf{e}(y'), N)$. The correctness and security is guaranteed by the forward and backward direction of Eq. (3), respectively. □

3 Pair Encoding Schemes

A pair encoding scheme (PES), introduced by Attrapadung [6], is an encoding system used in a general framework to construct ABE. Structures of a ciphertext and secret keys of an ABE scheme can be concisely captured by polynomials, and its decryption procedure can be represented by matrices. A PES is defined as a set of algorithms that output these polynomials or matrices. Intuitively, the polynomials specify the structures of exponent of group elements in a ciphertext and secret key, and the matrices specify coefficients used in the decryption.

3.1 Pair Encoding Scheme Definition

Definition 9 (Pair Encoding Schemes). Let $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ be a predicate family, indexed by $\kappa = (N, \mathsf{par})$, where par specifies some parameters. A PES for P_κ is given by four deterministic polynomial-time algorithms:

- $\mathsf{Param}(\mathsf{par}) \to \omega$. When given par as input, Param outputs $\omega \in \mathbb{N}$ that specifies the number of *common* variables, which we denote by $\mathbf{w} = (w_1, \ldots, w_\omega)$.
- $\mathsf{EncCt}(x, N) \to (n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}))$. On input $N \in \mathbb{N}$ and $x \in \mathcal{X}_{(N,\mathsf{par})}$, EncCt outputs a vector of polynomial $\mathbf{c} = (c_1, \ldots, c_{n_3})$ in *non-lone* variables $\mathbf{s} = (s_0, s_1, \ldots, s_{n_1})$ and *lone* variables $\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_{n_2})$ as follows, where $\theta_{i,z}, \theta_{i,t,j} \in \mathbb{Z}_N$:

$$\mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}) = \{ \sum_{z \in [n_2]} \theta_{i,z} \hat{s}_z + \sum_{t \in [n_1]^+, j \in [\omega]} \theta_{i,t,j} w_j s_t \}_{i \in [n_3]}.$$

- $\mathsf{EncKey}(y, N) \to (m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w}))$. On input $N \in \mathbb{N}$ and $y \in \mathcal{Y}_{(N,\mathsf{par})}$, EncKey outputs a vector of polynomial $\mathbf{k} = (k_1, \ldots, k_{m_3})$ in *non-lone* variables $\mathbf{r} = (r_1, \ldots, r_{m_1})$ and *lone* variables $\hat{\mathbf{r}} = (\alpha, \hat{r}_1, \ldots, \hat{r}_{m_2})$ as follows, where $\phi_i, \phi_{i,u}, \phi_{i,v,j} \in \mathbb{Z}_N$:

$$\mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w}) = \{ \phi_i \alpha + \sum_{u \in [m_2]} \phi_{i,u} \hat{r}_u + \sum_{v \in [m_1], j \in [\omega]} \phi_{i,v,j} w_j r_v \}_{i \in [m_3]}.$$

– $\mathsf{Pair}(x, y, N) \to (\mathbf{E}, \overline{\mathbf{E}})$. On input N, and both x, and y, Pair outputs two matrices $\mathbf{E}, \overline{\mathbf{E}}$ of sizes $(n_1+1) \times m_3$ and $n_3 \times m_1$, respectively.

Correctness A PES is said to be correct if for every $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $\mathsf{P}_\kappa(x, y) = 1$, then $\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \alpha s_0$ holds symbolically. The left-hand side is indeed a linear combination of $s_t k_p$ and $c_q r_v$, for $t \in [n_1]^+, p \in [m_3], q \in [n_3], v \in [m_1]$. Hence, an equivalent (and simpler) way to describe Pair and correctness together at once is to show such a linear combination that evaluates to αs_0.

Terminology We denote $(\hat{r}_1, \ldots, \hat{r}_{m_2})$ by $\hat{\mathbf{r}}_{-\alpha}$. Following [3], a variable is called *lone* as it is not multiplied with any w_j (otherwise called *non-lone*). Furthermore, since α, s_0 are treated distinguishably in defining correctness, we also often call them the *special* lone and non-lone variable, respectively. Throughout the paper, we fix N in index κ as prime p, which is an order of bilinear groups used to construct an ABE scheme. For notational conciseness, we consider that κ only specifies par, and p is hard-coded in EncCt, EncKey, and Pair.

3.2 Evaluating PES with Vectors/Matrices

We can evaluate ciphertext encoding $\mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})$ with the following substitution from scalar variables to vectors/matrices as follows. Let $d \in \mathbb{N}$. Each s_t is substituted by a vector $\mathbf{s}_t \in \mathbb{Z}_N^d$. Each $\hat{s}_z$ is substituted by a vector $\hat{\mathbf{s}}_z \in \mathbb{Z}_N^d$. Each w_j is substituted by a matrix $\mathbf{W}_j \in \mathbb{Z}_N^{d \times d}$. Let $\mathbf{S} = (\mathbf{s}_0, \ldots, \mathbf{s}_{n_1}) \in \mathbb{Z}_N^{d \times (n_1+1)}$, $\hat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2}) \in \mathbb{Z}_N^{d \times n_2}$, and $\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_\omega)$, we then define

$$\mathbf{c}(\mathbf{S}, \hat{\mathbf{S}}, \mathbb{W}) = \{ \sum_{z \in [n_2]} \theta_{i,z} \hat{\mathbf{s}}_z + \sum_{t \in [n_1]^+, j \in [\omega]} \theta_{i,t,j} \mathbf{W}_j^\top \mathbf{s}_t \}_{i \in [n_3]}.$$

Similarly for key encoding, let $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}) \in \mathbb{Z}_N^{d \times m_1}$, $\hat{\mathbf{R}} = (\mathbf{h}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2}) \in \mathbb{Z}_N^{d \times (m_2+1)}$, we then define

$$\mathbf{k}(\mathbf{R}, \hat{\mathbf{R}}, \mathbb{W}) = \{ \phi_i \mathbf{h} + \sum_{u \in [m_2]} \phi_{i,u} \hat{\mathbf{r}}_u + \sum_{v \in [m_1], j \in [\omega]} \phi_{i,v,j} \mathbf{W}_j \mathbf{r}_v \}_{i \in [m_3]}.$$

3.3 Security Properties of PESs

Definition 10 (Perfect Master-Key Hiding (PMH) [6]). Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be a PES for a predicate faimily $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0, 1\}$. We say that Γ satisfies perfect master-key hiding (PMH) if the following holds. Let $\omega \leftarrow \mathsf{Param}(\mathsf{par})$, $(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x)$, and $(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})) \leftarrow \mathsf{EncKey}(y)$. Then, for all κ and $(x, y) \in \mathcal{X}_\kappa \times \mathcal{Y}_\kappa$ such that $\mathsf{P}_\kappa(x, y) = 0$, the two distributions are identical, where the probability is taken over $\mathbf{s} \leftarrow \mathbb{Z}_p^{n_1+1}$, $\hat{\mathbf{s}} \leftarrow \mathbb{Z}_p^{n_2}$, $\mathbf{r} \leftarrow \mathbb{Z}_p^{m_1}$, $\alpha \leftarrow \mathbb{Z}_p$, $\hat{\mathbf{r}}_{-\alpha} \leftarrow \mathbb{Z}_p^{m_2}$, and $\mathbf{w} \leftarrow \mathbb{Z}_p^\omega$.

$$\{\mathbf{s}, \mathbf{r}, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}), \mathbf{k}(\mathbf{r}, (0, \hat{\mathbf{r}}_{-\alpha}), \mathbf{w})\} \quad \text{and} \quad \{\mathbf{s}, \mathbf{r}, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}), \mathbf{k}(\mathbf{r}, (\alpha, \hat{\mathbf{r}}_{-\alpha}), \mathbf{w})\}.$$

Definition 11 (Single-Variable PMH). We say that Γ satisfies single-variable PMH if Γ is PMH and $n_1 = 0$ for all $x \in \mathcal{X}_\kappa$, where $(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x)$. In other words, EncCt uses only s_0 for non-lone variable.

Note that Ambrona *et al.* showed that all predicate encodings [37] can be seen as a PES with single-variable PMH [5].

We next introduce the (ζ, ℓ)-key-encoding indistinguishability ((ζ, ℓ)-$\mathsf{KE\text{-}ind}$), which is a central security property in our framework, where we consider several transformations of PESs. The crucial feature on (ζ, ℓ)-$\mathsf{KE\text{-}ind}$ is two-fold: it is preserved after transformations, and it leads to the adaptive security of the resulting ABE scheme.

Definition 12 **((ζ,ℓ)-KE-ind).** Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be a PES for a predicate family $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$. Let $\zeta, \ell \in \mathbb{N}$ such that $\ell \le \zeta$. We say that Γ satisfies (ζ,ℓ)-KE-ind if the following holds. Consider a game $\mathsf{G}_\beta^{(\zeta,\ell)\text{-KE-ind}}$ defined in Fig 1, in which an adversary $\mathcal{A}$ can adaptively query $\mathcal{O}_\mathcal{X}$ and $\mathcal{O}_\mathcal{Y}$ with $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $\mathsf{P}_\kappa(x,y) = 0$, respectively. $\mathcal{A}$ is allowed to query each oracle at most once. Then, for all $\eta \in \{1,2\}$, we have $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}} \approx_c \mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}$.

Note that we can omit the terms that correspond to $g_{[1,i]}, h_{[1,i]}$ of the composite-order variant in the introduction by giving $\mathbf{a}_i^*, \mathbf{b}_i^*$ as $\mathbb{Z}_p$ elements to $\mathcal{A}$.

$\underline{\mathsf{G}_\beta^{(\zeta,\ell)\text{-KE-ind}}}$
$\omega \leftarrow \mathsf{Param}(\mathsf{par}),\ \ \mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$
$\overline{\mathbf{A}}, \overline{\mathbf{B}} \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)},\ \ \mathbb{W} = (\mathbf{W}_1, \dots, \mathbf{W}_\omega) \leftarrow (\mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)})^\omega$
$P = (\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}, \{[\mathbf{W}_i^\top \mathbf{A}]_\eta, [\mathbf{W}_i\mathbf{B}]_{3-\eta}\}_{i\in[\omega]})$
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{X}(\cdot), \mathcal{O}_\mathcal{Y}(\cdot,\cdot)}(P)$

$\underline{\mathcal{O}_\mathcal{X}(\cdot)}$
Input: $x \in \mathcal{X}_\kappa$
$(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x)$
$\mathbf{c}_0 \leftarrow \mathsf{span}(\mathbf{A}, \mathbf{a}_1, \dots, \mathbf{a}_\ell),\ \ \mathbf{s}_1, \dots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$
$\mathbf{S} = (\mathbf{c}_0, \mathbf{A}\mathbf{s}_1, \dots, \mathbf{A}\mathbf{s}_{n_1}),\ \ \widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{n_2})$
Output: $([\mathbf{S}]_\eta, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_\eta)$

$\underline{\mathcal{O}_\mathcal{Y}(\cdot,\cdot)}$
Input: $y \in \mathcal{Y}_\kappa$ and $\mathbf{h} \in \mathbb{Z}_p^{k+\zeta}$
$(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})) \leftarrow \mathsf{EncKey}(y)$
$\mu \leftarrow \mathbb{Z}_p,\ \ \mathbf{r}_1, \dots, \mathbf{r}_{m_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{r}}_1, \dots, \hat{\mathbf{r}}_{m_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$
$\mathbf{R} = (\mathbf{B}\mathbf{r}_1, \dots, \mathbf{B}\mathbf{r}_{m_1}),\ \ \widehat{\mathbf{R}} = (\mathbf{h} + \boxed{\beta\mu\mathbf{a}_\ell^*}, \hat{\mathbf{r}}_1, \dots, \hat{\mathbf{r}}_{m_2})$
Output: $([\mathbf{R}]_{3-\eta}, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_{3-\eta})$

Fig 1. (ζ,ℓ)-KE-ind game.

The following theorem says that all PESs with single-variable PMH satisfy (ζ,ℓ)-KE-ind for all $\zeta, \ell \in \mathbb{N}$.

Theorem 4 **((ζ,ℓ)-KE-ind of PES with Single-Variable PMH).** *Let Γ be a PES with single-variable PMH. Then, for all constants $\zeta, \ell \in \mathbb{N}$, Γ satisfies (ζ,ℓ)-KE-ind under the $\mathcal{D}_k$-MDDH assumption. More precisely, for all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}_{\mathcal{A},\Gamma}^{(\zeta,\ell)\text{-KE-ind}}(\lambda) \le 2\mathsf{Adv}_{\mathcal{B}}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. The proof of Theorem 4 is similar to the procedure that changes a normal secret key to a semi-functional one in the dual system methodology in prime-order groups [1,7,14]. Here, we follow the terminology by Chen et al. [14]. In the procedure, a normal key is first changed to a pseudo-normal one by a computational assumption. Then, it is changed to pseudo-semi-functional one by the information-theoretical security property of the encoding. Finally, it is changed to semi-functional one by a computational assumption.

We consider two hybrids H_1 and H_2 to prove the theorem. They are defined as follows:

H_1**:** Same as $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}$ except that $\mathbf{R} = (\mathbf{d}_1, \dots, \mathbf{d}_{m_1})$ where $\mathbf{d}_i \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1)$ in $\mathcal{O}_\mathcal{Y}$.
H_2**:** Same as $\mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}$ except that $\mathbf{R} = (\mathbf{d}_1, \dots, \mathbf{d}_{m_1})$ where $\mathbf{d}_i \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1)$ in $\mathcal{O}_\mathcal{Y}$.

We prove that $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}} \approx_c \mathsf{H}_1 \approx_s \mathsf{H}_2 \approx_c \mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}$. Intuitively, the output of $\mathcal{O}_\mathcal{Y}$ in $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}$, H_1, H_2, and $\mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}$ corresponds to a normal, pseudo-normal, pseudo-semi-functional, and semi-functional secret key, respectively. Thanks to Lemmata 4 to 6, Theorem 4 holds. $\square$

Lemma 4. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$|\Pr[\langle \mathcal{A}, \mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}\rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{H}_1\rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathcal{D}_k\text{-MDDH}}(\lambda).$$

Proof. We describe the reduction algorithm $\mathcal{B}$. $\mathcal{B}$ is given an instance of m_1-$\mathcal{D}_k$-MDDH problem. $(\mathbb{G}, [\mathbf{M}]_{3-\eta}, [\mathbf{T}_\beta]_{3-\eta})$ where $\mathbf{T}_0 = \mathbf{MU}$ and $\mathbf{T}_1 = \mathbf{V}$, where $\mathbf{U} \leftarrow \mathbb{Z}_p^{k\times m_1}$ and $\mathbf{V} \leftarrow \mathbb{Z}_p^{(k+1)\times m_1}$. Then, $\mathcal{B}$ chooses $\mathbf{X} \leftarrow \mathsf{GL}_{k+\zeta}(\mathbb{Z}_p)$ and sets

$$\overline{\mathbf{B}} = \mathbf{X}\begin{pmatrix} \widehat{\mathbf{M}} & & \\ \underline{\mathbf{M}} & 1 & \\ & & \mathbf{I}_{\zeta-1} \end{pmatrix},$$

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

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

$$[\mathbf{B}]_{3-\eta} = \left[\mathbf{X}\begin{pmatrix} \mathbf{M} \\ \mathbf{O} \end{pmatrix}\right]_{3-\eta}, \quad (\mathbf{b}_2^*||\dots||\mathbf{b}_\zeta^*) = (\mathbf{X}^\top)^{-1}\begin{pmatrix} \mathbf{O} \\ \mathbf{I}_{\zeta-1} \end{pmatrix}.$$

$\mathcal{B}$ generates $\overline{\mathbf{A}}$ and $\mathbb{W}$ by itself and computes the input P for $\mathcal{A}$ from them. When $\mathcal{A}$ queries $\mathcal{O}_{\mathcal{X}}$, $\mathcal{B}$ replies honestly as shown in Fig 1. Note that $\mathbf{S} = \mathbf{c}_0$ because EncCt uses only one non-lone variable. When $\mathcal{A}$ queries $\mathcal{O}_{\mathcal{Y}}$, $\mathcal{B}$ replies honestly except that it sets

$$[\mathbf{d}_i]_{3-\eta} = \left[\mathbf{X}\begin{pmatrix} \mathbf{t}_{\beta,i} \\ \mathbf{0} \end{pmatrix}\right]_{3-\eta}, \quad [\mathbf{R}]_{3-\eta} = [(\mathbf{d}_1, \dots, \mathbf{d}_{m_1})]_{3-\eta},$$

where $\mathbf{t}_{\beta,i}$ denotes the i-th column of $\mathbf{T}_\beta$. Because we can write

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

where $\mathbf{u}_i \leftarrow \mathbb{Z}_p^k$ and $u_i \leftarrow \mathbb{Z}_p$, $\mathbf{d}_i$ 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 $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}$ if $\beta = 0$, and H_1 otherwise. This concludes the proof. □

Lemma 5. *For all PPT adversaries $\mathcal{A}$, we have*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_1\rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{H}_2\rangle = 1]| \leq 2^{-\Omega(\lambda)}.$$

Proof. We redefine $\mathbf{W}_i = \widetilde{\mathbf{W}}_i + t_i \mathbf{a}_\ell^* {\mathbf{b}_1^*}^\top$ for $i \in [\omega]$, $\hat{\mathbf{s}}_i = \hat{\mathbf{s}}'_i + u_i \mathbf{b}_1^*$ for $i \in [n_2]$, and $\hat{\mathbf{r}}_i = \hat{\mathbf{r}}'_i + v_i \mathbf{a}_\ell^*$ for $i \in [m_2]$, where $\widetilde{\mathbf{W}}_i \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)}$, $\hat{\mathbf{s}}'_i, \hat{\mathbf{r}}'_i \leftarrow \mathbb{Z}_p^{k+\zeta}$, and $t_i, u_i, v_i \leftarrow \mathbb{Z}_p$. This change clearly does not affect the distributions of $\mathbf{W}_i$, $\hat{\mathbf{s}}_i$ and $\hat{\mathbf{r}}_i$. This affects $\mathcal{A}$'s view as follows:

P: $\mathbf{W}_i^\top \mathbf{A} = \widetilde{\mathbf{W}}_i^\top \mathbf{A}$, $\mathbf{W}_i \mathbf{B} = \widetilde{\mathbf{W}}_i \mathbf{B}$.

$\mathcal{O}_{\mathcal{X}}$: $\mathbf{c}(\mathbf{c}_0, \widehat{\mathbf{S}}, \mathbb{W}) = \mathbf{c}(\mathbf{c}_0, \widehat{\mathbf{S}}, \widetilde{\mathbb{W}}) + \mathbf{c}({\mathbf{a}_\ell^*}^\top \mathbf{c}_0, \mathbf{u}, \mathbf{t}) \otimes \mathbf{b}_1^*$, where $\widetilde{\mathbb{W}} = (\widetilde{\mathbf{W}}_1, \dots, \widetilde{\mathbf{W}}_\omega)$, $\mathbf{u} = (u_1, \dots, u_{n_2})$, and $\mathbf{t} = (t_1, \dots, t_\omega)$. Note that $\mathbf{c}({\mathbf{a}_\ell^*}^\top \mathbf{c}_0, \mathbf{u}, \mathbf{t}) \otimes \mathbf{b}_1^*$ denotes $(c_1 \mathbf{b}_1^*, \dots, c_{n_3} \mathbf{b}_1^*)$, where $(c_1, \dots, c_{n_3}) = \mathbf{c}({\mathbf{a}_\ell^*}^\top \mathbf{c}_0, \mathbf{u}, \mathbf{t})$.

$\mathcal{O}_{\mathcal{Y}}$: $\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W}) = \mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \widetilde{\mathbb{W}}) + \mathbf{k}(\mathbf{r}, \mathbf{v}, \mathbf{t}) \otimes \mathbf{a}_\ell^*$, where $\mathbf{r} = ({\mathbf{b}_1^*}^\top \mathbf{d}_1, \dots, {\mathbf{b}_1^*}^\top \mathbf{d}_{m_1})$ and $\mathbf{v} = (0, v_1, \dots, v_{m_2})$.

Thanks to the PMH of Γ, the following distributions are almost identical:

$$\{\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{c}_0, \mathbf{R}, \mathbf{c}({\mathbf{a}_\ell^*}^\top \mathbf{c}_0, \mathbf{u}, \mathbf{t}), \mathbf{k}(\mathbf{r}, \mathbf{v}, \mathbf{t})\} \qquad \text{and}$$
$$\{\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{c}_0, \mathbf{R}, \mathbf{c}({\mathbf{a}_\ell^*}^\top \mathbf{c}_0, \mathbf{u}, \mathbf{t}), \mathbf{k}(\mathbf{r}, \mathbf{v} + (\mu, \mathbf{0}), \mathbf{t})\},$$

where $\mu \leftarrow \mathbb{Z}_p$. This is because ${\mathbf{a}_\ell^*}^\top \mathbf{c}_0$ and ${\mathbf{b}_1^*}^\top \mathbf{d}_i$ are distributed statistically close to being uniform in $\mathbb{Z}_p$. Thus, $\mathcal{A}$'s view is not changed even if we change the first element of $\mathbf{v}$ from 0 to μ except negligible probability. In other words, $\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})$ and $\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W}) + \mathbf{k}(\mathbf{0}, (\mu, \mathbf{0}), \mathbf{0}) \otimes \mathbf{a}_\ell^*$ are identically distributed except negligible probability. The latter exactly corresponds to $\mathcal{A}$'s view in H_2. This concludes the proof. □

Lemma 6. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_2 \rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{G}_1^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} \rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}(\lambda).$$

This lemma can be proven similarly to Lemma 4.

4 Predicate Transformations

In this section, we present several transformations for predicates, which enable us to construct a more expressive predicate from simple predicates. As shown later in §6, these transformations are sufficiently powerful to construct ABE schemes whose constructions from standard assumptions are still unknown. Concretely, we introduce four transformations called the direct sum, dual transformation, KP augmentation, and CP augmentation. Because the CP augmentation is obtained from the dual transformation and KP augmentation, the former three transformations are sufficient for our framework. We also present the corresponding transformations of PESs for each predicate transformation and prove that these PES transformations preserve the (ζ, ℓ)-KE-ind property. Starting from PESs with the single-variable PMH, which already satisfy (ζ, ℓ)-KE-ind, we can obtain a PES for a expressive predicate that satisfies (ζ', ζ')-KE-ind for some constant ζ'. Finally, we show that we can use the PES with (ζ', ζ')-KE-ind to construct an adaptively secure ABE scheme in §5.

4.1 Direct Sum of Predicate Families

Definition 13 (Direct Sum [8]). Let $\mathsf{P}_{\kappa_i}^{(i)} : \mathcal{X}_{\kappa_i}^{(i)} \times \mathcal{Y}_{\kappa_i}^{(i)} \to \{0,1\}$ be a predicate family. Let $\kappa = (\kappa_1, \ldots, \kappa_d)$. A predicate family for the direct sum of a predicate family set $\mathcal{P}_\kappa = (\mathsf{P}_{\kappa_1}^{(1)}, \ldots, \mathsf{P}_{\kappa_d}^{(d)})$, denoted by $\mathsf{DS}[\mathcal{P}_\kappa] : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{0,1\}$, is defined as follows: let $\bar{\mathcal{X}}_\kappa = \bigcup_{i\in[d]}(\{i\} \times \mathcal{X}_{\kappa_i}^{(i)})$, $\bar{\mathcal{Y}}_\kappa = \bigcup_{i\in[d]}(\{i\} \times \mathcal{Y}_{\kappa_i}^{(i)})$, and define

$$\mathsf{DS}[\mathcal{P}_\kappa]((i_x, x), (i_y, y)) \Leftrightarrow (i_x = i_y) \wedge (\mathsf{P}_{\kappa_{i_y}}^{(i_y)}(x, y) = 1).$$

We sometimes use another notation, $\mathsf{P}_{\kappa_1}^{(1)} \odot \cdots \odot \mathsf{P}_{\kappa_d}^{(d)}$, to denotes $\mathsf{DS}[\mathcal{P}_\kappa]$.

PES for $\mathsf{DS}[\mathcal{P}_\kappa]$. Let $\Gamma_i = (\mathsf{Param}_i, \mathsf{EncCt}_i, \mathsf{EncKey}_i, \mathsf{Pair}_i)$ be a PES for $\mathsf{P}_{\kappa_i}^{(i)}$. We construct a PES for $\mathsf{DS}[\mathcal{P}_\kappa]$, denoted by $\mathsf{DS\text{-}Trans}(\mathbf{\Gamma}) = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$, where $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_d)$.

- $\mathsf{Param}'(\mathsf{par}) \to \omega'$: Run $\omega_i \leftarrow \mathsf{Param}_i(\mathsf{par})$ and output $\sum_{i\in[d]} \omega_i$. This specifies common variables $\mathbf{w}' = (\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(d)})$, where $\mathbf{w}^{(i)} = (w_1^{(i)}, \ldots, w_{\omega_i}^{(i)})$.
- $\mathsf{EncCt}'((i_x, x)) \to (n_1', n_2', \mathbf{c}'(\mathbf{s}', \hat{\mathbf{s}}', \mathbf{w}'))$:
  - Output $(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}^{(i_x)})) \leftarrow \mathsf{EncCt}_{i_x}(x)$.
  - Define $n_1' = n_1$, $n_2' = n_2$, $\mathbf{s}' = \mathbf{s}$, and $\hat{\mathbf{s}}' = \hat{\mathbf{s}}$.

– $\mathsf{EncKey}'((i_y, y)) \to (m'_1, m'_2, \mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}'))$:
  - Output $(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w}^{(i_y)})) \leftarrow \mathsf{EncKey}_{i_y}(y)$.
  - Define $m'_1 = m_1$, $m'_2 = m_2$, $\mathbf{r}' = \mathbf{r}$, and $\hat{\mathbf{r}}' = \hat{\mathbf{r}}$.

– $\mathsf{Pair}'((i_x, x), (i_y, y)) \to (\mathbf{E}', \overline{\mathbf{E}}')$ and correctness:
  - Output $(\mathbf{E}, \overline{\mathbf{E}}) \leftarrow \mathsf{Pair}_{i_y}(x, y)$.
  - Correctness of Pair' directly follows from that of Pair_{i_y}.

Theorem 5 ((ζ,ℓ)-KE-ind of DS-Trans($\mathbf{\Gamma}$)). *If Γ_i satisfies (ζ,ℓ)-KE-ind for all $i \in [d]$, then DS-Trans($\mathbf{\Gamma}$) satisfies (ζ,ℓ)-KE-ind. More precisely, for all PPT adversaries $\mathcal{A}$, there exist PPT adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{(\zeta,\ell)\text{-KE-ind}}_{\mathcal{A},\mathsf{DS\text{-}Trans}(\mathbf{\Gamma})}(\lambda) \le d \max_{i\in[d]} \mathsf{Adv}^{(\zeta,\ell)\text{-KE-ind}}_{\mathcal{B},\Gamma_i}(\lambda).$$

$\underline{\mathsf{G}^{(\zeta,\ell)\text{-KE-ind}}_{\beta}}$
$\omega_i \leftarrow \mathsf{Param}_i(\mathsf{par}),\ \ \mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$
$\overline{\mathbf{A}}, \overline{\mathbf{B}} \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)},\ \ \mathbb{W}_i = (\mathbf{W}_{i,1}, \ldots, \mathbf{W}_{i,\omega_i}) \leftarrow (\mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)})^{\omega_i}$
$P = (\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}, \{[\mathbf{W}_{i,j}^\top \mathbf{A}]_\eta, [\mathbf{W}_{i,j}\mathbf{B}]_{3-\eta}\}_{i\in[d], j\in[\omega_i]})$
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\bar{\mathcal{X}}}(\cdot), \mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}(P)$

$\underline{\mathcal{O}_{\bar{\mathcal{X}}}(\cdot)}$
Input: $(i_x, x) \in \bar{\mathcal{X}}_\kappa$
$(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}^{(i_x)})) \leftarrow \mathsf{EncCt}_{i_x}(x)$
$\mathbf{c}_0 \leftarrow \mathsf{span}(\mathbf{A}, \mathbf{a}_1, \ldots, \mathbf{a}_\ell),\ \ \mathbf{s}_1, \ldots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$
$\mathbf{S} = (\mathbf{c}_0, \mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{n_1}),\ \ \widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2})$
Output: $([\mathbf{S}]_\eta, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W}_{i_x})]_\eta)$

$\underline{\mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}$
Input: $(i_y, y) \in \bar{\mathcal{Y}}_\kappa$ and $\mathbf{h} \in \mathbb{Z}_p^{k+\zeta}$
$(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w}^{(i_y)})) \leftarrow \mathsf{EncKey}_{i_y}(y)$
$\mu \leftarrow \mathbb{Z}_p,\ \ \mathbf{r}_1, \ldots, \mathbf{r}_{m_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$
$\mathbf{R} = (\mathbf{B}\mathbf{r}_1, \ldots, \mathbf{B}\mathbf{r}_{m_1}),\ \ \widehat{\mathbf{R}} = (\mathbf{h} + \boxed{\beta\mu\mathbf{a}_\ell^*}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2})$
Output: $([\mathbf{R}]_{3-\eta}, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W}_{i_y})]_{3-\eta})$

Fig 2. (ζ,ℓ)-KE-ind game for DS-Trans($\mathbf{\Gamma}$).

Proof. For $\beta \in \{0,1\}$, we can describe the (ζ,ℓ)-KE-ind game $\mathsf{G}^{(\zeta,\ell)\text{-KE-ind}}_{\beta}$ for DS-Trans($\mathbf{\Gamma}$) as shown in Fig 2. To prove the theorem, we consider an adversary $\mathcal{B}$, which samples $t \leftarrow [d]$ and interacts with $\mathcal{O}_{\mathcal{X}^{(t)}}$ and $\mathcal{O}_{\mathcal{Y}^{(t)}}$ of the (ζ,ℓ)-KE-ind game for Γ_t. $\mathcal{B}$ internally runs an adversary $\mathcal{A}$ against (ζ,ℓ)-KE-ind of DS-Trans($\mathbf{\Gamma}$) and interacts with it as follows:

1. Let $\omega_i \leftarrow \mathsf{Param}_i(\mathsf{par})$. $\mathcal{B}$ is given $(\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}, \{[\mathbf{W}_{t,j}^\top \mathbf{A}]_\eta, [\mathbf{W}_{t,j}\mathbf{B}]_{3-\eta}\}_{j\in[\omega_t]})$. It then samples $\mathbb{W}_i = (\mathbf{W}_{i,1}, \ldots, \mathbf{W}_{i,\omega_i}) \leftarrow (\mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)})^{\omega_i}$ for $i \in [d]\backslash t$.
2. $\mathcal{B}$ gives to $\mathcal{A}$ the following elements: $\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}$, together with $\{[\mathbf{W}_{i,j}^\top \mathbf{A}]_\eta, [\mathbf{W}_{i,j}\mathbf{B}]_{3-\eta}\}_{i\in[d], j\in[\omega_i]}$
3. For $\mathcal{A}$'s query to $\mathcal{O}_{\bar{\mathcal{X}}}$ on (i_x, x), $\mathcal{B}$ replies as follows:
   – If $i_x = t$, $\mathcal{B}$ queries its own oracle $\mathcal{O}_{\mathcal{X}^{(t)}}$ on x and gives the reply, $([\mathbf{S}]_\eta, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W}_t)]_\eta)$, to $\mathcal{A}$.

– If $i_x \neq t$, $\mathcal{B}$ computes $\mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}^{(i_x)}), \mathbf{S}$, and $\widehat{\mathbf{S}}$ as show below, and gives $([\mathbf{S}]_\eta, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W}_{i_x})]_\eta)$ to $\mathcal{A}$:

$$
\begin{aligned}
&(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w}^{(i_x)})) \leftarrow \mathsf{EncCt}_{i_x}(x), \quad \mathbf{c}_0 \leftarrow \mathsf{Ker}(\mathbf{a}^*_{\ell+1}, \ldots, \mathbf{a}^*_\zeta), \\
&\mathbf{s}_1, \ldots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k, \quad \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta} \\
&\mathbf{S} = (\mathbf{c}_0, \mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{n_1}), \quad \widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2}).
\end{aligned}
$$

Note that $\mathsf{span}(\mathbf{A}, \mathbf{a}_1, \ldots, \mathbf{a}_\ell) = \mathsf{Ker}(\mathbf{a}^*_{\ell+1}, \ldots, \mathbf{a}^*_\zeta)$.

4. For $\mathcal{A}$'s query to $\mathcal{O}_{\bar{\mathcal{Y}}}$ on (i_y, y), $\mathcal{B}$ replies as follows:
   – If $i_y = t$, $\mathcal{B}$ queries its own oracle $\mathcal{O}_{\mathcal{Y}^{(t)}}$ on y and gives the reply, which is $([\mathbf{R}]_{3-\eta}, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W}_t)]_{3-\eta})$, to $\mathcal{A}$. Note that the first element of $\widehat{\mathbf{R}}$ is $\mathbf{h}$ (if $\beta = 0$) or $\mathbf{h} + \mu\mathbf{a}^*_\ell$ (if $\beta = 1$).
   – If $i_y \neq t$, $\mathcal{B}$ aborts the interaction with $\mathcal{A}$ and outputs a random bit $\beta' \leftarrow \{0, 1\}$.
5. $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is.

In the above experiment, $\mathcal{B}$ correctly simulates $\mathcal{O}_{\bar{\mathcal{X}}}$. Since $\mathcal{B}$ aborts the experiment if $i_y \neq t$, we focus on the case of $i_y = t$, which occurs with probability $1/d$. Note that since $i_x = t \Rightarrow \mathsf{P}^{(t)}(x, y) = 0$ from the game condition for $\mathsf{DS\text{-}Trans}(\mathbf{\Gamma})$, $\mathcal{B}$ follow the game condition for Γ_t. If $\beta = 0$ in the $\mathsf{KE\text{-}ind}$ game for Γ_t, $\mathcal{A}$'s view corresponds to that in $\mathsf{G}_0^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$, and it corresponds to $\mathsf{G}_1^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$ otherwise. Thus, we have $\Pr[i_y = t] \cdot \mathsf{Adv}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}_{\mathcal{A},\mathsf{DS\text{-}Trans}(\mathbf{\Gamma})}(\lambda) + \Pr[i_y \neq t] \cdot 0 \leq \mathsf{Adv}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}_{\mathcal{B},\Gamma_t}(\lambda) \leq \max_{i \in [d]} \mathsf{Adv}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}_{\mathcal{B},\Gamma_i}(\lambda)$. This concludes the proof. □

4.2 Dual Predicates

Recall that the dual of $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0, 1\}$ is $\mathsf{Dual}[\mathsf{P}_\kappa] : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{0, 1\}$ where $\bar{\mathcal{X}}_\kappa = \mathcal{Y}_\kappa$ and $\bar{\mathcal{Y}}_\kappa = \mathcal{X}_\kappa$, and $\mathsf{Dual}[\mathsf{P}_\kappa](x, y) = \mathsf{P}_\kappa(y, x)$.

PES for $\mathsf{Dual}[\mathsf{P}_\kappa]$. Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be a PES for P_κ. We construct a PES for $\mathsf{Dual}[\mathsf{P}_\kappa]$, denoted by $\mathsf{Dual\text{-}Trans}(\Gamma) = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$.

– $\mathsf{Param}'(\mathsf{par}) \to \omega'$: Run $\omega \leftarrow \mathsf{Param}(\mathsf{par})$ and output $\omega + 1$. This specifies common variables $\mathbf{w}' = (w_0, w_1, \ldots, w_\omega)$, where w_0 is a new common variable.
– $\mathsf{EncCt}'(x) \to (n'_1, n'_2, \mathbf{c}'(\mathbf{s}', \hat{\mathbf{s}}', \mathbf{w}'))$:
  • Run $(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})) \leftarrow \mathsf{EncKey}(x)$. Let s_{new} be a new special non-lone variable. Polynomials $\mathbf{c}'(\mathbf{s}', \hat{\mathbf{s}}', \mathbf{w}')$ are defined the same as $\mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})$ except that α is replaced with $s_{\mathsf{new}} w_0$.
  • Define $n'_1 = m_1$, $n'_2 = m_2$, $\mathbf{s}' = (s_{\mathsf{new}}, \mathbf{r})$, and $\hat{\mathbf{s}}' = \hat{\mathbf{r}}_{-\alpha}$.
– $\mathsf{EncKey}'(y) \to (m'_1, m'_2, \mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}))$:
  • Run $(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(y)$. Let α_{new} be a new special lone variable. Polynomials $\mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}')$ are defined the same as $\mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})$ except that a polynomial $\alpha_{\mathsf{new}} - s_0 w_0$ is added as the first element of $\mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}')$.
  • Define $m'_1 = n_1 + 1$, $m'_2 = n_2$, $\mathbf{r}' = \mathbf{s}$, and $\hat{\mathbf{r}}' = (\alpha_{\mathsf{new}}, \hat{\mathbf{s}})$.
– $\mathsf{Pair}'(x, y) \to (\mathbf{E}', \overline{\mathbf{E}}')$ and correctness:
  • Run $(\mathbf{E}, \overline{\mathbf{E}}) \leftarrow \mathsf{Pair}(y, x)$. Define $\mathbf{E}' = \begin{pmatrix} 1 & \\ & \overline{\mathbf{E}}^\top \end{pmatrix}$ and $\overline{\mathbf{E}}' = \mathbf{E}^\top$.
  • For correctness, we have

$$
\begin{aligned}
\mathbf{s}'\mathbf{E}'\mathbf{k}'^\top + \mathbf{c}'\overline{\mathbf{E}}'\mathbf{r}'^\top =& (s_{\mathsf{new}}, \mathbf{r}) \begin{pmatrix} 1 & \\ & \overline{\mathbf{E}}^\top \end{pmatrix} (\alpha_{\mathsf{new}} - s_0 w_0, \mathbf{c})^\top + \mathbf{k}|_{\alpha \mapsto s_{\mathsf{new}} w_0} \mathbf{E}^\top \mathbf{s}^\top \\
=& s_{\mathsf{new}}\alpha_{\mathsf{new}} - s_{\mathsf{new}} s_0 w_0 + s_{\mathsf{new}} s_0 w_0 = s_{\mathsf{new}}\alpha_{\mathsf{new}}.
\end{aligned}
$$

Theorem 6 ((ζ, ℓ)-KE-ind of $\mathsf{Dual\text{-}Trans}(\Gamma)$). *Let $2 \leq \ell \leq \zeta$. If Γ satisfies $(\zeta, \ell - 1)$-*$\mathsf{KE\text{-}ind}$*, then* $\mathsf{Dual\text{-}Trans}(\Gamma)$ *satisfies (ζ, ℓ)-*$\mathsf{KE\text{-}ind}$ *under the $\mathcal{D}_k$-MDDH assumption. More precisely, for all PPT adversaries $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$
\mathsf{Adv}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}_{\mathcal{A},\mathsf{Dual\text{-}Trans}(\Gamma)}(\lambda) \leq \mathsf{Adv}^{(\zeta,\ell-1)\text{-}\mathsf{KE\text{-}ind}}_{\mathcal{B}_1,\Gamma}(\lambda) + 2\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)}.
$$

Proof. For $\beta \in \{0,1\}$, we can describe the (ζ,ℓ)-KE-ind game $\mathsf{G}_\beta^{(\zeta,\ell)\text{-KE-ind}}$ for $\mathsf{Dual\text{-}Trans}(\Gamma)$ as shown in Fig 3. To show this theorem, we consider two intermediate hybrids H_1 and H_2, which are also described in Fig 3. That is, H_1 (resp. H_2) is defined the same as $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}$ (resp. $\mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}$) except that $\mathbf{d}_0$, the first elements of $\mathbf{R}$ generated in $\mathcal{O}_{\bar{\mathcal{Y}}}$, is set as $\mathbf{d}_0 \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1, \ldots, \mathbf{b}_{\ell-1})$ instead of $\mathbf{B}\mathbf{r}_0$ where $\mathbf{r}_0 \leftarrow \mathbb{Z}_p^k$. Then, we prove that $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}} \approx_c \mathsf{H}_1 \approx_c \mathsf{H}_2 \approx_c \mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}$. Thanks to Lemmata 7 to 9, we can see that these hybrids are indistinguishable, and Theorem 6 holds. □

$\mathsf{G} \in \left\{ \mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}, \boxed{\mathsf{H}_1}, \boxed{\boxed{\mathsf{H}_2}, \mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}} \right\}$

$\underline{\mathsf{G}}$
$\omega \leftarrow \mathsf{Param}(\mathsf{par}),\ \ \mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$
$\overline{\mathbf{A}}, \overline{\mathbf{B}} \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)},\ \ \mathbb{W} = (\mathbf{W}_0^\top, \mathbf{W}_1^\top, \ldots, \mathbf{W}_\omega^\top) \leftarrow (\mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)})^{\omega+1}$
$P = (\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}, \{[\mathbf{W}_i^\top \mathbf{A}]_\eta, [\mathbf{W}_i \mathbf{B}]_{3-\eta}\}_{i\in[\omega]^+})$
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\bar{\mathcal{X}}}(\cdot), \mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}(P)$

$\underline{\mathcal{O}_{\bar{\mathcal{X}}}(\cdot)}$
Input: $x \in \bar{\mathcal{X}}_\kappa$
$(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})) \leftarrow \mathsf{EncKey}(x)$
$\mathbf{c}_0 \leftarrow \mathsf{span}(\mathbf{A}, \mathbf{a}_1, \ldots, \mathbf{a}_\ell),\ \ \mathbf{s}_1, \ldots, \mathbf{s}_{m_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{m_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$
$\mathbf{S} = (\mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{m_1}),\ \ \widehat{\mathbf{S}} = (\mathbf{W}_0^\top \mathbf{c}_0, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{m_2})$
Output: $([\mathbf{c}_0]_\eta, [\mathbf{S}]_\eta, [\mathbf{k}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_\eta)$

$\underline{\mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}$
Input: $y \in \bar{\mathcal{Y}}_\kappa$ and $\mathbf{h} \in \mathbb{Z}_p^{k+\zeta}$
$(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(y)$
$\mu \leftarrow \mathbb{Z}_p,\ \ \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{n_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$
$\mathbf{d}_0 = \mathbf{B}\mathbf{r}_0,\ \ \boxed{\mathbf{d}_0 \leftarrow \mathsf{span}(\mathbf{B}, \mathbf{b}_1, \ldots, \mathbf{b}_{\ell-1})}$
$\mathbf{R} = (\mathbf{d}_0, \mathbf{B}\mathbf{r}_1, \ldots, \mathbf{B}\mathbf{r}_{n_1}),\ \ \widehat{\mathbf{R}} = (\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{n_2})$
Output: $([\mathbf{h} + \boxed{\mu \mathbf{a}_\ell^*} - \mathbf{W}_0 \mathbf{d}_0]_{3-\eta}, [\mathbf{R}]_{3-\eta}, [\mathbf{c}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_{3-\eta})$

Fig 3. (ζ,ℓ)-KE-ind game for $\mathsf{Dual\text{-}Trans}(\Gamma)$.

Lemma 7. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\langle \mathcal{A}, \mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}} \rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{H}_1 \rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathcal{D}_k\text{-MDDH}}(\lambda)$.

Proof. We describe the reduction algorithm $\mathcal{B}$. $\mathcal{B}$ is given an instance of $\mathcal{U}_{k+\ell-1,k}$ problem, $(\mathbb{G}, [\mathbf{M}]_{3-\eta}, [\mathbf{t}_\beta]_{3-\eta})$ where $\mathbf{t}_0 = \mathbf{M}\mathbf{u}$ and $\mathbf{t}_1 = \mathbf{v}$, where $\mathbf{u} \leftarrow \mathbb{Z}_p^k$ and $\mathbf{v} \leftarrow \mathbb{Z}_p^{k+\ell-1}$. Then, $\mathcal{B}$ chooses $\mathbf{X} \leftarrow \mathsf{GL}_{k+\zeta}(\mathbb{Z}_p)$ and sets

$$\overline{\mathbf{B}} = \mathbf{X} \begin{pmatrix} \widehat{\mathbf{M}} & & \\ \underline{\mathbf{M}} & \mathbf{I}_{\ell-1} & \\ & & \mathbf{I}_{\zeta-\ell+1} \end{pmatrix},$$

$$(\overline{\mathbf{B}}^\top)^{-1} = (\mathbf{X}^\top)^{-1} \begin{pmatrix} (\widehat{\mathbf{M}}^\top)^{-1} & -(\widehat{\mathbf{M}}^\top)^{-1}\underline{\mathbf{M}}^\top & \\ & \mathbf{I}_{\ell-1} & \\ & & \mathbf{I}_{\zeta-\ell+1} \end{pmatrix},$$

where $\widehat{\mathbf{M}}$ is the matrix consisting of the first k rows of $\mathbf{M}$, and $\underline{\mathbf{M}}$ is that consisting of the last $\ell-1$ rows of $\mathbf{M}$. Then, $\mathcal{B}$ can compute

$$[\mathbf{B}]_{3-\eta} = \left[\mathbf{X}\begin{pmatrix} \mathbf{M} \\ \mathbf{O} \end{pmatrix}\right]_{3-\eta},\quad (\mathbf{b}_{\ell+1}^* || \ldots || \mathbf{b}_\zeta^*) = (\mathbf{X}^\top)^{-1}\begin{pmatrix} \mathbf{O} \\ \mathbf{I}_{\zeta-\ell} \end{pmatrix}.$$

$\mathcal{B}$ generates $\overline{\mathbf{A}}$ and $\mathbb{W}$ by itself and computes the input P for $\mathcal{A}$ from them. When $\mathcal{A}$ queries $\mathcal{O}_{\bar{\mathcal{X}}}$, $\mathcal{B}$ replies honestly as shown in Fig 3. When $\mathcal{A}$ queries $\mathcal{O}_{\bar{\mathcal{Y}}}$, $\mathcal{B}$ replies honestly except that it sets

$$[\mathbf{d}_0]_{3-\eta} = \left[\mathbf{X}\begin{pmatrix}\mathbf{t}_\beta\\ \mathbf{0}\end{pmatrix}\right]_{3-\eta}, \quad [\mathbf{R}]_{3-\eta} = [(\mathbf{d}_0, \mathbf{B}\mathbf{r}_1, \dots, \mathbf{B}\mathbf{r}_{m_1})]_{3-\eta}.$$

Now since we can write $\mathbf{t}_\beta = \begin{pmatrix}\widehat{\mathbf{M}}\\ \underline{\mathbf{M}}\end{pmatrix}\mathbf{u}_1 + \beta\begin{pmatrix}\mathbf{O}\\ \mathbf{I}_{\ell-1}\end{pmatrix}\mathbf{u}_2$, where $\mathbf{u}_1 \leftarrow \mathbb{Z}_p^k$ and $\mathbf{u}_2 \leftarrow \mathbb{Z}_p^{\ell-1}$, we have that $\mathbf{d}_0$ is uniformly distributed in $\mathsf{span}(\mathbf{B})$ if $\beta = 0$, and in $\mathsf{span}(\mathbf{B}, \mathbf{b}_1, \dots, \mathbf{b}_{\ell-1})$ otherwise. Thus, the view of $\mathcal{A}$ corresponds to $\mathsf{G}_0^{(\zeta,\ell)\text{-KE-ind}}$ if $\beta = 0$, and H_1 otherwise. This concludes the proof. □

Lemma 8. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$|\Pr[\langle\mathcal{A}, \mathsf{H}_1\rangle = 1] - \Pr[\langle\mathcal{A}, \mathsf{H}_2\rangle = 1]| \le \mathsf{Adv}_{\mathcal{B},\Gamma}^{(\zeta,\ell-1)\text{-KE-ind}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that the outputs of $\mathcal{O}_{\bar{\mathcal{Y}}}$ in H_1 and H_2 are computationally indistinguishable if the PES Γ for P_κ satisfies $(\zeta, \ell-1)$-KE-ind. We construct a PPT adversary $\mathcal{B}$ against $(\zeta, \ell-1)$-KE-ind of Γ that internally runs a PPT distinguisher $\mathcal{A}$ between H_1 and H_2. $\mathcal{B}$ behaves as follows.

1. $\mathcal{B}$ is given an input of $(\zeta, \ell-1)$-KE-ind game for Γ, $(\mathbb{G}, [\mathbf{M}]_{3-\eta}, [\mathbf{N}]_\eta, \{\mathbf{m}_i^*\}_{i\in[\ell-1,\zeta]}, \{\mathbf{n}_i^*\}_{i\in[\ell,\zeta]}, \{[\mathbf{V}_i^\top\mathbf{M}]_{3-\eta}, [\mathbf{V}_i\mathbf{N}]_\eta\}_{i\in[\omega]})$. $\mathcal{B}$ implicitly defines that $\mathbf{A} = \mathbf{N}$, $\mathbf{B} = \mathbf{M}$, and $\mathbf{W}_i = \mathbf{V}_i^\top$ for $i \in [\omega]$.
2. $\mathcal{B}$ samples $\mathbf{W}_0 \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)}$ and gives $P = (\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}, \{[\mathbf{W}_i^\top\mathbf{A}]_\eta, [\mathbf{W}_i\mathbf{B}]_{3-\eta}\}_{i\in[\omega]^+})$ to $\mathcal{A}$.
3. For $\mathcal{A}$'s query to $\mathcal{O}_{\bar{\mathcal{X}}}$ on x, $\mathcal{B}$ samples $\mathbf{c}_0 \leftarrow \mathsf{Ker}(\mathbf{a}_{\ell+1}^*, \dots, \mathbf{a}_\zeta^*)$ and queries its own oracle $\mathcal{O}_{\mathcal{Y}}$ on $(x, \mathbf{W}_0^\top\mathbf{c}_0)$ to obtain $([\mathbf{T}]_\eta, [\mathbf{k}(\mathbf{T}, \widehat{\mathbf{T}}, \mathbb{V})]_\eta)$, where
$$\begin{aligned}\mathbf{T} &= (\mathbf{N}\mathbf{t}_0, \mathbf{N}\mathbf{t}_1, \dots, \mathbf{N}\mathbf{t}_{m_1}) = (\mathbf{A}\mathbf{t}_0, \mathbf{A}\mathbf{t}_1, \dots, \mathbf{A}\mathbf{t}_{m_1}),\\ \widehat{\mathbf{T}} &= (\mathbf{W}_0^\top\mathbf{c}_0 + \beta\hat{\mu}\mathbf{m}_{\ell-1}^*, \hat{\mathbf{t}}_1, \dots, \hat{\mathbf{t}}_{m_2}) = (\mathbf{W}_0^\top\mathbf{c}_0 + \beta\hat{\mu}\mathbf{b}_{\ell-1}^*, \hat{\mathbf{t}}_1, \dots, \hat{\mathbf{t}}_{m_2}),\\ \mathbb{V} &= (\mathbf{V}_1, \dots, \mathbf{V}_\omega) = (\mathbf{W}_1^\top, \dots, \mathbf{W}_\omega^\top).\end{aligned}$$
Note that $\hat{\mu}$ is a random value in $\mathbb{Z}_p$ chosen by $\mathcal{O}_{\mathcal{Y}}$. $\mathcal{B}$ implicitly defines that $\mathbf{s}_i = \mathbf{t}_i$ for $i \in [m_1]^+$, $\hat{\mathbf{s}}_i = \hat{\mathbf{t}}_i$ for $i \in [m_2]$, $\mathbf{S} = \mathbf{T}$, $\widehat{\mathbf{S}} = \widehat{\mathbf{T}}$, and $\mathbb{W} = \mathbb{V}$. $\mathcal{B}$ replies $([\mathbf{c}_0]_\eta, [\mathbf{S}]_\eta, [\mathbf{k}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_\eta)$ to $\mathcal{A}$. Note that $\mathsf{span}(\mathbf{A}, \mathbf{a}_1, \dots, \mathbf{a}_\ell) = \mathsf{Ker}(\mathbf{a}_{\ell+1}^*, \dots, \mathbf{a}_\zeta^*)$.
4. For $\mathcal{A}$'s query to $\mathcal{O}_{\bar{\mathcal{Y}}}$ with y and $\mathbf{h}$, $\mathcal{B}$ queries its own oracle $\mathcal{O}_{\mathcal{X}}$ on y to obtain $([\mathbf{U}]_{3-\eta}, [\mathbf{c}(\mathbf{U}, \widehat{\mathbf{U}}, \mathbb{V})]_{3-\eta})$, where
$$\mathbf{U} = (\mathbf{o}_0, \mathbf{M}\mathbf{u}_1, \dots, \mathbf{M}\mathbf{u}_{n_1}) = (\mathbf{o}_0, \mathbf{B}\mathbf{u}_1, \dots, \mathbf{B}\mathbf{u}_{n_1}), \quad \widehat{\mathbf{U}} = (\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{n_2}).$$
Note that $\mathbf{o}_0$ is randomly distributed in $\mathsf{span}(\mathbf{M}, \mathbf{m}_1, \dots, \mathbf{m}_{\ell-1})$, which equals to $\mathsf{span}(\mathbf{B}, \mathbf{b}_1, \dots, \mathbf{b}_{\ell-1})$. $\mathcal{B}$ implicitly defines that $\mathbf{r}_i = \mathbf{u}_i$ for $i \in [n_1]$, $\hat{\mathbf{r}}_i = \hat{\mathbf{u}}_i$ for $i \in [n_2]$, $\mathbf{R} = \mathbf{U}$, $\widehat{\mathbf{R}} = \widehat{\mathbf{U}}$, and $\mathbf{d}_0 = \mathbf{o}_0$. $\mathcal{B}$ replies $([\mathbf{h} - \mathbf{W}_0\mathbf{d}_0]_{3-\eta}, [\mathbf{R}]_{3-\eta}, [\mathbf{c}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_{3-\eta})$ to $\mathcal{A}$.
5. $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is.

At a glance, this simulation seems that the distribution of the reply from $\mathcal{O}_{\bar{\mathcal{X}}}$ is changed. However, entire views of $\mathcal{A}$ correspond to H_1 and H_2. To see this, we redefine $\mathbf{W}_0$ as $\mathbf{W}_0 = \widetilde{\mathbf{W}}_0 - \frac{\beta\hat{\mu}}{\mathbf{a}_\ell^{*\top}\mathbf{c}_0}\mathbf{a}_\ell^*\mathbf{b}_{\ell-1}^{*\top}$ where $\widetilde{\mathbf{W}}_0 \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)}$. Clearly, this does not change the distribution of $\mathbf{W}_0$. This affects $\mathcal{A}$'s view as follows:

$$\begin{aligned}P &: \ \mathbf{W}_0^\top\mathbf{A} = \widetilde{\mathbf{W}}_0^\top\mathbf{A}, \ \ \mathbf{W}_0\mathbf{B} = \widetilde{\mathbf{W}}_0\mathbf{B}.\\ \mathcal{O}_{\bar{\mathcal{X}}} &: \ \mathbf{W}_0^\top\mathbf{c}_0 + \beta\hat{\mu}\mathbf{b}_{\ell-1}^* = \widetilde{\mathbf{W}}_0^\top\mathbf{c}_0.\\ \mathcal{O}_{\bar{\mathcal{Y}}} &: \ \mathbf{h} - \mathbf{W}_0\mathbf{d}_0 = \mathbf{h} - \widetilde{\mathbf{W}}_0\mathbf{d}_0 + \frac{\beta\hat{\mu}\mathbf{b}_{\ell-1}^{*\top}\mathbf{d}_0}{\mathbf{a}_\ell^{*\top}\mathbf{c}_0}\mathbf{a}_\ell^* = \mathbf{h} - \widetilde{\mathbf{W}}_0\mathbf{d}_0 + \beta\mu\mathbf{a}_\ell^*.\end{aligned}$$

Because $\hat{\mu}$ is randomly distributed in $\mathbb{Z}_p$, we can set $\mu = \frac{\hat{\mu}\mathbf{b}_{\ell-1}^{*\top}\mathbf{d}_0}{\mathbf{a}_\ell^{*\top}\mathbf{c}_0}$ if $\mathbf{b}_{\ell-1}^{*\top}\mathbf{d}_0 \neq 0$ and $\mathbf{a}_\ell^{*\top}\mathbf{c}_0 \neq 0$. Since $\mathbf{c}_0$ and $\mathbf{d}_0$ are randomly distributed in $\mathsf{span}(\mathbf{A}, \mathbf{a}_1, \ldots, \mathbf{a}_\ell)$ and $\mathsf{span}(\mathbf{B}, \mathbf{b}_1, \ldots, \mathbf{b}_{\ell-1})$, respectively, this is the case with an overwhelming probability. Thus, $\mathcal{A}$'s view corresponds to H_1 if $\beta = 0$ in the (ζ, ℓ)-KE-ind game of Γ, and it corresponds to H_2 otherwise. This concludes the proof. □

Lemma 9. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that* $|\Pr[\langle \mathcal{A}, \mathsf{H}_2\rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{G}_1^{(\zeta,\ell)\text{-KE-ind}}\rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathcal{D}_k\text{-MDDH}}(\lambda)$.

The proof of Lemma 9 is similar to Lemma 7, and hence we omit it here.

4.3 Key-Policy Augmentation

Definition 14 (Key-Policy Augmentation). A predicate family for key-policy Boolean formula augmentation over a single predicate family $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$, denoted by $\mathsf{KBF1}[\mathsf{P}_\kappa] : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{0,1\}$, is defined as follows:

- $\bar{\mathcal{X}}_\kappa = \mathcal{X}_\kappa$ and $\bar{\mathcal{Y}}_\kappa = \bigcup_{i\in\mathbb{N}}(\mathcal{Y}_\kappa^i \times \mathcal{F}_i)$, where $\mathcal{F}_i$ consists of all monotone Boolean formulae with input length i.
- For $x \in \bar{\mathcal{X}}_\kappa$ and $y = ((y_1, \ldots, y_n), f) \in \bar{\mathcal{Y}}_\kappa$ where $f : \{0,1\}^n \to \{0,1\}$, we define $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ as $b_i = \mathsf{P}_\kappa(x, y_i)$. Then, $\mathsf{KBF1}[\mathsf{P}_\kappa](x, y) = 1 \Leftrightarrow f(b) = 1$.

We use $\mathsf{KBF1}_{\mathsf{OR}}[\mathsf{P}_\kappa]$ (resp. $\mathsf{KBF1}_{\mathsf{AND}}[\mathsf{P}_\kappa]$) to denote a predicate family that is the same as $\mathsf{KBF1}[\mathsf{P}_\kappa]$ except that $\mathcal{F}_i$ in $\bar{\mathcal{Y}}_\kappa$ consists of monotone Boolean formulae whose all gates are OR (resp. AND) gates. Note that the number 1 in $\mathsf{KBF1}$ refers to the property that the augmentation is over *one* predicate family. An augmentation over a *set* of predicate families follows similarly in analogous manner to [8], and we defer to §6 (and §6.2). Following [8], the term *dynamic* composition refers to the property that f can be chosen freely as a part of key attribute (as opposed to static compositions, where f is fixed once and for all). The term *unbounded* composition refers to the property that the input length n of f is unbounded (see a more precise definition when we define the augmentation over sets in §6.2).

PES for $\mathsf{KBF1}[\mathsf{P}_\kappa]$. Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be a PES for P_κ. We construct a PES for $\mathsf{KBF1}[\mathsf{P}_\kappa]$, denoted by $\mathsf{KBF1\text{-}Trans}(\Gamma) = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$. Let $\mathsf{Share_p}$ be the linear secret sharing algorithm on polynomials defined in Fig 4.

$\mathsf{Share_p}(f, \alpha, \mathbf{u})$

Input: A monotone Boolean formula $f = (n, w, m, G)$ with τ AND gates, a variable α, and τ variables $\mathbf{u} = (u_1, \ldots, u_\tau)$.

1. Set a variable $\sigma_{\mathsf{out}} = \alpha$ on the output wire.
2. Let $\phi : [m] \to [\tau]$ be a function such that $\phi(i) = |\{j \mid j \leq i \wedge G_1(j) = \text{AND}\}|$, where $G_1(j)$ denotes the first element of $G(j)$. For each AND gate g with incoming wires a, b and an outgoing wire c where a polynomial σ_c is set on c, set new polynomials $\sigma_a = \sigma_c - u_{\phi(g)}$ and $\sigma_b = u_{\phi(g)}$ on a and b, respectively.
3. For each OR gate g with incoming wires a, b and an outgoing wire c where a vector σ_c is set on c, set new polynomials $\sigma_a = \sigma_c$ and $\sigma_b = \sigma_c$ on a and b, respectively.
4. Output polynomials $\sigma_1, \ldots, \sigma_n$, which are set on the input wires $1, \ldots, n$.

Fig 4. Linear secret sharing scheme for Boolean formulae on polynomials.

- $\mathsf{Param}'(\mathsf{par}) = \mathsf{Param}(\mathsf{par})$ and $\mathsf{EncCt}'(x) = \mathsf{EncCt}(x)$
- $\mathsf{EncKey}'((y_1, \ldots, y_n), f) \to (m_1', m_2', \mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}))$:
  - For $i \in [n]$, run $\mathsf{EncKey}(y_i)$ to obtain n sets of polynomials $\mathbf{k}^{(1)}, \ldots, \mathbf{k}^{(n)}$, where $\mathbf{k}^{(i)} = \mathbf{k}(\mathbf{r}^{(i)}, \hat{\mathbf{r}}^{(i)}, \mathbf{w})$.

- Let τ be a number of AND gates in f. Let α_{new} be a new special lone variable and $\mathbf{u} = (u_1, \ldots, u_\tau)$ be new lone variables. Let $\sigma_1, \ldots, \sigma_n$ be polynomials that are an output of $\mathsf{Share}_{\mathsf{p}}(f, \alpha_{\mathsf{new}}, \mathbf{u})$. A new set of polynomials $\mathbf{k}'^{(i)}$ is defined the same as $\mathbf{k}^{(i)}$ except that the variable $\alpha^{(i)}$ in each polynomial is replaced with σ_i.
- Define $m_1' = nm_1$, $m_2' = \tau + nm_2$, and $\mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{w}) = (\mathbf{k}'^{(1)}, \ldots, \mathbf{k}'^{(n)})$. Note that $\mathbf{r}' = (\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(n)})$ and $\hat{\mathbf{r}}' = (\alpha_{\mathsf{new}}, \mathbf{u}, \hat{\mathbf{r}}^{(1)}_{-\alpha^{(1)}}, \ldots, \hat{\mathbf{r}}^{(n)}_{-\alpha^{(n)}})$.

– $\mathsf{Pair}'(x, y) \to (\mathbf{E}', \overline{\mathbf{E}}')$ and correctness:
  - Let polynomials $\sigma_1, \ldots, \sigma_n$ be an output of $\mathsf{Share}_{\mathsf{p}}(f, \alpha_{\mathsf{new}}, \mathbf{u})$. It is not hard to see that, for all $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ such that $f(b) = 1$, there exists a set $S \subseteq \{i \mid b_i = 1\}$ such that $\sum_{i \in S} \sigma_i = \alpha_{\mathsf{new}}$. Thus, if x and $y = ((y_1, \ldots, y_n), f)$ satisfy $\mathsf{KBF1}[\mathsf{P}_\kappa](x, y) = 1$, there exists $S \subseteq \{i \mid \mathsf{P}_\kappa(x, y_i) = 1\}$ such that $\sum_{i \in S} \sigma_i = \alpha_{\mathsf{new}}$.
  - For $i \in S$, run $\mathsf{Pair}(x, y_i)$ to obtain $(\mathbf{E}^{(i)}, \overline{\mathbf{E}}^{(i)})$, which satisfies $\mathbf{s}\mathbf{E}^{(i)}\mathbf{k}^{(i)^\top} + \mathbf{c}\overline{\mathbf{E}}^{(i)}\mathbf{r}^{(i)^\top} = \sigma_i s_0$. Then, we can obtain $\sum_{i \in S} \sigma_i s_0 = \alpha_{\mathsf{new}} s_0$ by the linear combination.

Theorem 7 ((ζ, ℓ)-KE-ind of KBF1-Trans($\mathit{\Gamma}$)). *Let B be the maximum depth of f chosen by $\mathcal{A}$ in the (ζ, ℓ)-KE-ind game for $\mathsf{KBF1}$-$\mathsf{Trans}(\mathit{\Gamma})$. If $\mathit{\Gamma}$ satisfies (ζ, ℓ)-KE-ind, then $\mathsf{KBF1}$-$\mathsf{Trans}(\mathit{\Gamma})$ satisfies (ζ, ℓ)-KE-ind as long as $B = O(\log \lambda)$. That is, for all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}_{\mathcal{A},\mathsf{KBF1}\text{-}\mathsf{Trans}(\mathit{\Gamma})}(\lambda) \leq 2^{9B+1}\mathsf{Adv}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}_{\mathcal{B},\mathit{\Gamma}}(\lambda).$$

We prove Theorem 7 by extending the techniques regarding pebbling arguments that Kowalczyk-Wee [25] have introduced in proving adaptive security of their ABE schemes for formulae with multi-use.

Proof. We utilize the piecewise guessing framework (Section 2.3) by Kowalczyk and Wee [25] to prove Theorem 7. However, they use a secret sharing scheme that puts shares on all nodes of Boolean formula, whereas our transformation puts shares on only input nodes Fig 4 and 5. Very recently, Tomida et al. presented an improved technique that allows us to use the piecewise guessing framework with shares on only input nodes [34]. Hence, we proceed the proof following their improved strategy.

Let $\mathsf{G}_0^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ and $\mathsf{G}_1^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ be the (ζ, ℓ)-KE-ind games for $\mathsf{KBF1}$-$\mathsf{Trans}(\mathit{\Gamma})$. First, we define a linear secret sharing algorithm Share on vectors as shown in Fig 5. Then, we have Lemma 10 for the secret sharing scheme.

$\mathsf{Share}(f, \mathbf{z})$

Input: A monotone Boolean formula $f = (n, w, m, G)$ and a secret $\mathbf{z} \in \mathbb{Z}_p^\gamma$, where γ is arbitrary natural number.

1. Set a vector $\boldsymbol{\sigma}_{\mathsf{out}} = \mathbf{z}$ 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 $\mathbf{u}_g \leftarrow \mathbb{Z}_p^\gamma$ 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 $1, \ldots, n$.

Fig 5. Linear secret sharing scheme for Boolean formulae.

Lemma 10. *For all $\gamma, n \in \mathbb{N}$, monotone Boolean formulae $f = (n, w, m, G)$, $\mathbf{h}, \mathbf{a} \in \mathbb{Z}_p^\gamma$, and $\mu \in \mathbb{Z}_p$, we define the following distribution.*

$$\mathbf{h}_1, \ldots, \mathbf{h}_n \leftarrow \mathsf{Share}(f, \mathbf{h} + \mu\mathbf{a}), \quad \mathbf{h}_1', \ldots, \mathbf{h}_n' \leftarrow \mathsf{Share}(f, \mathbf{h}),$$
$$\mu_1, \ldots, \mu_n \leftarrow \mathsf{Share}(f, \mu).$$

Then, the two distributions are identical:

$$\{\mathbf{h}_1, \ldots, \mathbf{h}_n\} \text{ and } \{\mathbf{h}'_1 + \mu_1\mathbf{a}, \ldots, \mathbf{h}'_n + \mu_n\mathbf{a}\}.$$

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 $\mathbf{h}_i$, $\mathbf{h}'_i$, and μ_i for $i \in [w]$ be values set on wire i in the execution of $\mathsf{Share}(f, \mathbf{h}+\mu\mathbf{a})$, $\mathsf{Share}(f, \mathbf{h})$, and $\mathsf{Share}(f, \mu)$, respectively. Then, we have

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

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 $\mathbf{h}_i = \mathbf{h}'_i + \mu_i\mathbf{a}$ for $i \in [w]$. This concludes the proof. $\square$

We can describe $\mathsf{G}_\beta^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$ for $\beta \in \{0, 1\}$ using the secret sharing algorithm Share as shown in Fig 6. By applying Lemma 10, $\mathbf{h}_i$ in $\mathcal{O}_{\bar{\mathcal{Y}}}$ can be replaced with $\mathbf{h}'_i + \mu_i\mathbf{a}_1^*$ where $\mathbf{h}'_1, \ldots, \mathbf{h}'_n \leftarrow \mathsf{Share}(f, \mathbf{h})$ and $\mu_1, \ldots, \mu_n \leftarrow \mathsf{Share}(f, \beta\mu)$. In what follows, we use the latter definition for $\mathcal{O}_{\bar{\mathcal{Y}}}$.

$\underline{\mathsf{G}_\beta^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}}$ $\omega \leftarrow \mathsf{Param}(\mathsf{par})$, $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$ $\overline{\mathbf{A}}, \overline{\mathbf{B}} \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)}$, $\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_\omega) \leftarrow (\mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)})^\omega$ $P = (\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i\in[\ell,\zeta]}, \{\mathbf{b}_i^*\}_{i\in[\ell+1,\zeta]}, \{[\mathbf{W}_i^\top\mathbf{A}]_\eta, [\mathbf{W}_i\mathbf{B}]_{3-\eta}\}_{i\in[\omega]})$ $\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\bar{\mathcal{X}}}(\cdot), \mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}(P)$
$\underline{\mathcal{O}_{\bar{\mathcal{X}}}(\cdot)}$ Input: $x \in \bar{\mathcal{X}}_\kappa$ $(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x)$ $\mathbf{c}_0 \leftarrow \mathsf{span}(\mathbf{A}, \mathbf{a}_1, \ldots, \mathbf{a}_\ell)$, $\mathbf{s}_1, \ldots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k$, $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$ $\mathbf{S} = (\mathbf{c}_0, \mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{n_1})$, $\widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2})$ Output: $([\mathbf{S}]_\eta, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_\eta)$
$\underline{\mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}$ Input: $y = ((y_1, \ldots, y_n), f) \in \bar{\mathcal{Y}}_\kappa$ and $\mathbf{h} \in \mathbb{Z}_p^{k+\zeta}$ $(m_1, m_2, \mathbf{k}(\mathbf{r}^{(i)}, \hat{\mathbf{r}}^{(i)}, \mathbf{w})) \leftarrow \mathsf{EncKey}(y_i)$ $\mu \leftarrow \mathbb{Z}_p$, $\mathbf{r}_1^{(i)}, \ldots, \mathbf{r}_{m_1}^{(i)} \leftarrow \mathbb{Z}_p^k$, $\hat{\mathbf{r}}_1^{(i)}, \ldots, \hat{\mathbf{r}}_{m_2}^{(i)} \leftarrow \mathbb{Z}_p^{k+\zeta}$ $\underbrace{\mathbf{h}_1, \ldots, \mathbf{h}_n \leftarrow \mathsf{Share}(f, \mathbf{h} + \boxed{\beta\mu\mathbf{a}_\ell^*})}_{\substack{\mathbf{h}'_1,\ldots,\mathbf{h}'_n \leftarrow \mathsf{Share}(f,\mathbf{h}), \\ \mu_1,\ldots,\mu_n \leftarrow \mathsf{Share}(f, \boxed{\beta\mu}) \\ \mathbf{h}_i = \mathbf{h}'_i + \mu_i\mathbf{a}_\ell^*}}$ $\mathbf{R}^{(i)} = (\mathbf{B}\mathbf{r}_1^{(i)}, \ldots, \mathbf{B}\mathbf{r}_{m_1}^{(i)})$, $\widehat{\mathbf{R}}^{(i)} = (\mathbf{h}_i, \hat{\mathbf{r}}_1^{(i)}, \ldots, \hat{\mathbf{r}}_{m_2}^{(i)})$ Output: $\{([\mathbf{R}^{(i)}]_{3-\eta}, [\mathbf{k}(\mathbf{R}^{(i)}, \widehat{\mathbf{R}}^{(i)}, \mathbb{W})]_{3-\eta})\}_{i\in[n]}$

Fig 6. (ζ, ℓ)-$\mathsf{KE\text{-}ind}$ game for $\mathsf{KBF1\text{-}Trans}(\Gamma)$.

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 $\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$ and $\mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$, which satisfy

- $\widehat{\mathsf{G}}_0^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}} = \widehat{\mathsf{H}}^{h_0} \approx_c, \ldots, \approx_c \widehat{\mathsf{H}}^{h_L} = \widehat{\mathsf{G}}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$
- $\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}} = \mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE\text{-}ind}}$.

The function h_ι for $\iota \in [L]$ is defined as follows. Let $z = (x, y) \in \{0,1\}^R$ on which $\mathcal{A}$ queries $\mathcal{O}_{\bar{\mathcal{X}}}$ and $\mathcal{O}_{\bar{\mathcal{Y}}}$, respectively. Let $b \in \{0,1\}^n$ be a string computed from z following Definition 14. Note that $f(b) = 0$ because the game imposes the condition $\mathsf{KBF1}[\mathsf{P}_\kappa](x, y) = 0$ on $\mathcal{A}$. Let $\mathcal{R} = (r_1, \ldots, r_L) = \mathsf{PebRec}(f, b)$ be a pebbling record generated as shown in Lemma 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 part $\mathsf{Share}(f, \beta\mu)$ in $\mathcal{O}_{\bar{\mathcal{Y}}}$ from $\mathsf{G}_\beta^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$. That is, $\widehat{\mathsf{H}}^{h_\iota}$ is the same as $\widehat{\mathsf{G}}_0^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ except that $\mathsf{Share}(f, 0)$ is replaced with $\widetilde{\mathsf{Share}}(f, 0, h_\iota(z))$. The description of algorithm $\widetilde{\mathsf{Share}}$ is presented in Fig 7. $\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ is the same as H^{h_L}, and $\mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ is the same as $\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ except that $\widetilde{\mathsf{Share}}(f, 0, h_L(z))$ is replaced with $\widetilde{\mathsf{Share}}(f, \mu, h_L(z))$. The behaviors of $\mathcal{O}_{\bar{\mathcal{Y}}}$ in these hybrids are summarized in Fig 8.

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

Input: A monotone Boolean formula $f = (n, w, m, G)$ with depth $d \leq B$, $\mathbf{z} \in \mathbb{Z}_p^\gamma$, and $u \in \{0,1\}^{3B}$.

1. Set a vector $\boldsymbol{\sigma}_{\mathsf{out}} = \mathbf{z}$ 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 $\mathbf{u}_{g,1}, \mathbf{u}_{g,2} \leftarrow \mathbb{Z}_p^\gamma$ 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 $\mathbf{u}_g \leftarrow \mathbb{Z}_p^\gamma$ 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 $\mathbf{u}_i \leftarrow \mathbb{Z}_p^k$, i.e., $\boldsymbol{\sigma}_i = \mathbf{u}_i$.
7. Output shares $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n$, which are set on the input wires $1, \ldots, n$.

Fig 7. Description of $\widetilde{\mathsf{Share}}$.

$$\mathsf{G} \in \left\{ \mathsf{G}_0^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}, \boxed{\widehat{\mathsf{H}}^{h_\iota}}, \boxed{\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}}, \boxed{\mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}} \right\}$$

$\underline{\mathcal{O}_{\bar{\mathcal{Y}}}(\cdot,\cdot)}$

Input: $y = ((y_1, \ldots, y_n), f) \in \bar{\mathcal{Y}}_\kappa$ and $\mathbf{h} \in \mathbb{Z}_p^{k+\zeta}$

$(m_1, m_2, \mathbf{k}(\mathbf{r}^{(i)}, \hat{\mathbf{r}}^{(i)}, \mathbf{w})) \leftarrow \mathsf{EncKey}(y_i)$

$\mu \leftarrow \mathbb{Z}_p$, $\mathbf{r}_1^{(i)}, \ldots, \mathbf{r}_{m_1}^{(i)} \leftarrow \mathbb{Z}_p^k$, $\hat{\mathbf{r}}_1^{(i)}, \ldots, \hat{\mathbf{r}}_{m_2}^{(i)} \leftarrow \mathbb{Z}_p^{k+\zeta}$, $\mathbf{h}'_1, \ldots, \mathbf{h}'_n \leftarrow \mathsf{Share}(f, \mathbf{h})$

$\mu_1, \ldots, \mu_n \leftarrow \mathsf{Share}(f, 0)$, $\boxed{\mu_1, \ldots, \mu_n \leftarrow \widetilde{\mathsf{Share}}(f, 0, h_\iota(z))}$

$\boxed{\mu_1, \ldots, \mu_n \leftarrow \widetilde{\mathsf{Share}}(f, 0, h_L(z))}$, $\boxed{\mu_1, \ldots, \mu_n \leftarrow \widetilde{\mathsf{Share}}(f, \mu, h_L(z))}$

$\mathbf{h}_i = \mathbf{h}'_i + \mu_i \mathbf{a}_\ell^*$

$\mathbf{R}^{(i)} = (\mathbf{B}\mathbf{r}_1^{(i)}, \ldots, \mathbf{B}\mathbf{r}_{m_1}^{(i)})$, $\widehat{\mathbf{R}}^{(i)} = (\mathbf{h}_i, \hat{\mathbf{r}}_1^{(i)}, \ldots, \hat{\mathbf{r}}_{m_2}^{(i)})$

Output: $\{([\mathbf{R}^{(i)}]_{3-\eta}, [\mathbf{k}(\mathbf{R}^{(i)}, \widehat{\mathbf{R}}^{(i)}, \mathbb{W})]_{3-\eta})\}_{i\in[n]}$

Fig 8. Description of $\mathcal{O}_{\bar{\mathcal{Y}}}$ in hybrids.

We prove that

- $\mathsf{G}_0^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} \approx_c \mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$,
- $\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} = \mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$,
- $\mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} \approx_c \mathsf{G}_1^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$.

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.

$\underline{\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} = \mathsf{G}_{\mathsf{M1}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}}$**.** 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 7). 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.

$\underline{\mathsf{G}_{0}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} \approx_c \mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}}$**.** Following Lemma 1, we prove the two properties:

1. $\mathsf{G}_{0}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}} = \mathsf{H}^{h_0}$ and $\mathsf{H}^{h_L} = \mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$,
2. $\widehat{\mathsf{H}}_1^{h_{\iota-1}} \approx_c \widehat{\mathsf{H}}_0^{h_\iota}$ for $\iota \in [L]$.

where $\widehat{\mathsf{H}}_\beta^{h_i}$ for $\beta \in \{0,1\}$ is defined in Section 2.3. For item 1, the latter holds because we defined $\mathsf{G}_{\mathsf{M0}}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}$ 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 7, and the remaining procedures are exactly the same as $\mathsf{Share}(f, 0)$. Thus, the former also holds.

The remaining thing is to prove $\widehat{\mathsf{H}}_1^{h_{\iota-1}} \approx_c \widehat{\mathsf{H}}_0^{h_\iota}$. Formally, we prove Lemma 11, which allows us to conclude the proof of Theorem 7 from the observation so far and Lemma 1. □

Lemma 11. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$|\Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_1^{h_{\iota-1}} \rangle = 1] - \Pr[\langle \mathcal{A}, \widehat{\mathsf{H}}_0^{h_\iota} \rangle = 1]| \leq \mathsf{Adv}_{\mathcal{B},\Gamma}^{(\zeta,\ell)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}(\lambda).$$

Proof. 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 6. Recall that the difference between $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ is the input of $\widetilde{\mathsf{Share}}$. That is, the input is $(f, 0, u_0)$ in $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $(f, 0, u_1)$ in $\widehat{\mathsf{H}}_0^{h_\iota}$. Thus, in case of $u_0 = u_1$, $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are clearly identical. In the following, we consider the case of $u_0 \neq u_1$.

Let an object O be either a gate g^* with incoming wires a, b and an outgoing c or an input wire i^* in which the difference between $C(f, u_0)$ and $C(f, u_1)$ lies. In what follows, we only consider the case where a pebble is placed on O when we move from $C(f, u_0)$ to $C(f, u_1)$. Note that we can similarly analyze the opposite case, where a pebble is removed, by just considering the reverse of the former case. When O is a gate g^*, $\widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\widetilde{\mathsf{Share}}(f, 0, u_1)$ are identically distributed, and thus $\widehat{\mathsf{H}}_1^{h_{\iota-1}}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are identical. We explain the reason in the following.

Consider the case where g^* is an AND gate 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'. This follows from the pebbling rule. Without loss of generality, 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 the procedure for 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 7). 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 7). Thus, the outputs of $\widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\widetilde{\mathsf{Share}}(f, 0, u_1)$ are identically distributed. We can easily observe that similar argument holds if g^* is an OR gate.

The remaining case is when O is an input wire $i^* \in [n]$. In this case, we show that the outputs of $\mathcal{O}_{\bar{\mathcal{Y}}}$ in $\widehat{\mathsf{H}}_1^{h_\iota - 1}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$ are computationally indistinguishable if the PES Γ for P_κ satisfies (ζ, ℓ)-KE-ind. The difference between these games is that μ_{i^*} is exactly one of the output of $\widetilde{\mathsf{Share}}(f, 0, u_0)$ in $\widehat{\mathsf{H}}_1^{h_\iota - 1}$, whereas μ_{i^*} is a random elements in $\mathbb{Z}_p$ in $\widehat{\mathsf{H}}_0^{h_\iota}$. This is because a pebble is not placed on i^* in the configuration $C(f, u_0)$ but is placed on i^* in the configuration $C(f, u_1)$. Thus, the output of $\widetilde{\mathsf{Share}}(f, 0, u_1)$ is the same as that of $\widetilde{\mathsf{Share}}(f, 0, u_0)$ except that μ_{i^*} is replaced with a random element (see item 6 in Fig 7).

We construct a PPT adversary $\mathcal{B}$ against (ζ, ℓ)-KE-ind of Γ that internally runs a PPT distinguisher $\mathcal{A}$ between $\widehat{\mathsf{H}}_1^{h_\iota - 1}$ and $\widehat{\mathsf{H}}_0^{h_\iota}$. $\mathcal{B}$ behaves as follows.

1. $\mathcal{A}$ commits (u_0, u_1) to $\mathcal{B}$.
2. $\mathcal{B}$ is given a parameter $(\mathbb{G}, [\mathbf{A}]_\eta, [\mathbf{B}]_{3-\eta}, \{\mathbf{a}_i^*\}_{i \in [\ell, \zeta]}, \{\mathbf{b}_i^*\}_{i \in [\ell+1, \zeta]}, \{[\mathbf{W}_i^\top \mathbf{A}]_\eta, [\mathbf{W}_i \mathbf{B}]_{3-\eta}\}_{i \in [\omega]})$ from the (ζ, ℓ)-KE-ind game for Γ and gives it to $\mathcal{A}$ as its input.
3. For $\mathcal{A}$'s query to $\mathcal{O}_{\bar{\mathcal{X}}}$, $\mathcal{B}$ just relays the query to its own oracle $\mathcal{O}_{\mathcal{X}}$ and the reply to $\mathcal{A}$.
4. For $\mathcal{A}$'s query to $\mathcal{O}_{\bar{\mathcal{Y}}}$ on $y = ((y_1, \ldots, y_n), f)$ and $\mathbf{h}$, $\mathcal{B}$ computes $\mu_1, \ldots, \mu_n \leftarrow \widetilde{\mathsf{Share}}(f, 0, u_0)$ and $\mathbf{h}_i$ for all $i \in [n]$ using $\mathbf{a}_\ell^*$ as shown in Fig 8. Then, $\mathcal{B}$ queries its own oracle $\mathcal{O}_{\mathcal{Y}}$ on y_{i^*} and $\mathbf{h}_{i^*}$ and obtains $[\mathbf{R}^{(i^*)}]_{3-\eta}$ and $[\mathbf{k}(\mathbf{R}^{(i^*)}, \widehat{\mathbf{R}}^{(i^*)}, \mathbb{W})]_{3-\eta}$. $\mathcal{B}$ also computes all terms in $\mathbf{R}^{(i)} = (\mathbf{B}\mathbf{r}_1^{(i)}, \ldots, \mathbf{B}\mathbf{r}_{m_1}^{(i)})$ and $\widehat{\mathbf{R}}^{(i)} = (\mathbf{h}_i, \hat{\mathbf{r}}_1^{(i)}, \ldots, \hat{\mathbf{r}}_{m_2}^{(i)})$ for all $i \in [n] \backslash i^*$ and $\{[\mathbf{k}(\mathbf{R}^{(i)}, \widehat{\mathbf{R}}^{(i)}, \mathbb{W})]_{3-\eta}\}_{i \in [n] \backslash i^*}$ by itself. $\mathcal{B}$ replies $\{([\mathbf{R}^{(i)}]_{3-\eta}, [\mathbf{k}(\mathbf{R}^{(i)}, \widehat{\mathbf{R}}^{(i)}, \mathbb{W})]_{3-\eta})\}_{i \in [n]}$ to $\mathcal{A}$.
5. $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is.

Observe that if $\beta = 0$ in the (ζ, ℓ)-KE-ind game of Γ, the first element of $\widehat{\mathbf{R}}^{(i^*)}$ is $\mathbf{h}_{i^*}$, and thus $\mathcal{A}$'s view corresponds to $\widehat{\mathsf{H}}_1^{h_\iota - 1}$. On the other hand, if $\beta = 1$ in the (ζ, ℓ)-KE-ind game of Γ, the first element of $\widehat{\mathbf{R}}^{(i^*)}$ is $\mathbf{h}_{i^*} + \hat{\mu}\mathbf{a}_\ell^* = \mathbf{h}'_{i^*} + \mu_{i^*}\mathbf{a}_\ell^* + \hat{\mu}\mathbf{a}_\ell^*$, where $\hat{\mu}$ is a random element in $\mathbb{Z}_p$ chosen by $\mathcal{O}_{\mathcal{Y}}$. It means that μ_{i^*} is randomized by $\hat{\mu}$, and $\mathcal{A}$'s view corresponds to $\widehat{\mathsf{H}}_0^{h_\iota}$. This concludes the proof. $\square$

Ciphertext-Policy Augmentation. Analogously to [8], for a predicate family P, we define its CP augmentation predicate—denoted as $\mathsf{CBF1}[\mathsf{P}]$—as the dual of $\mathsf{KBF1}[\mathsf{P}']$ where P' is the dual of P. Therefore, we can use the dual conversion—applying two times–sandwiching $\mathsf{KBF1}$-Trans, to obtain a PES conversion for $\mathsf{CBF1}[\mathsf{P}]$.

Definition 15 (Ciphertext-Policy Augmentation). A predicate family for ciphertext-policy Boolean formula augmentation of a predicate family $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0, 1\}$, denoted by $\mathsf{CBF1}[\mathsf{P}_\kappa] : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{0, 1\}$, is define as follows:

- $\bar{\mathcal{X}}_\kappa = \bigcup_{i \in \mathbb{N}} (\mathcal{X}_\kappa^i \times \mathcal{F}_i)$, where $\mathcal{F}_i$ consists of all monotone Boolean formulae with input length i.
- $\bar{\mathcal{Y}}_\kappa = \mathcal{Y}_\kappa$.
- For $x = ((x_1, \ldots, x_n), f) \in \bar{\mathcal{X}}_\kappa$ and $y \in \bar{\mathcal{Y}}_\kappa$ where $f : \{0,1\}^n \to \{0,1\}$, we define $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ as $b_i = \mathsf{P}_\kappa(x_i, y)$. Then, $\mathsf{CBF1}[\mathsf{P}_\kappa](x, y) = 1 \Leftrightarrow f(b) = 1$.

Additionally, $\mathsf{CBF1}_{\mathsf{OR}}[\mathsf{P}_\kappa]$ and $\mathsf{CBF1}_{\mathsf{AND}}[\mathsf{P}_\kappa]$ are defined in the same way as KP augmentation.

PES for $\mathsf{CBF1}[\mathsf{P}_\kappa]$. Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be a PES for P_κ. It is not hard to see that $\mathsf{CBF1}[\mathsf{P}_\kappa] = \mathsf{Dual}[\mathsf{KBF1}[\mathsf{Dual}[\mathsf{P}_\kappa]]]$. Therefore, we can obtain a PES for $\mathsf{CBF1}[\mathsf{P}_\kappa]$, denoted by $\mathsf{CBF1}$-$\mathsf{Trans}(\Gamma) = (\mathsf{Param}', \mathsf{EncCt}', \mathsf{EncKey}', \mathsf{Pair}')$, from Dual-Trans and $\mathsf{KBF1}$-Trans. That is,

$$\mathsf{CBF1}\text{-}\mathsf{Trans}(\Gamma) = \mathsf{Dual}\text{-}\mathsf{Trans}(\mathsf{KBF1}\text{-}\mathsf{Trans}(\mathsf{Dual}\text{-}\mathsf{Trans}(\Gamma))).$$

4.4 Conforming PES for ABE

We can apply our transformations, namely, direct sum, dual, and key-policy augmentation, to a predicate family set $\mathcal{P}_\kappa$ multiple times to obtain a new predicate family P_κ. When we apply a PES to construct an ABE scheme, (ζ', ζ')-KE-ind for some constant ζ' implies the adaptive security of the resulting ABE scheme. The following theorem says that if we have predicate families $\mathcal{P}_\kappa = (\mathsf{P}^{(1)}_{\kappa_1}, \ldots, \mathsf{P}^{(d)}_{\kappa_d})$ that satisfy (ζ, ℓ)-KE-ind for all constants $\ell, \zeta \in \mathbb{N}$, we can construct an ABE scheme for a predicate family P_κ obtained by applying the above transformations to $\mathcal{P}_\kappa$ arbitrarily many times.

To state the theorem formally, we define a composed predicate set $f_c(\mathcal{P}_\kappa)$ for a predicate family set $\mathcal{P}_\kappa = (\mathsf{P}^{(1)}_{\kappa_1}, \ldots, \mathsf{P}^{(d)}_{\kappa_d})$. Let $\bar{\mathcal{P}}_\kappa$ be a predicate family set that consists of all predicate families obtained by applying one of transformations, $(\mathsf{DS}, \mathsf{Dual}, \mathsf{KBF1})$, to $\mathcal{P}_\kappa$. That is, $\bar{\mathcal{P}}_\kappa = (\mathsf{DS}[\mathcal{P}_\kappa], \{\mathsf{Dual}[\mathsf{P}^{(i)}_{\kappa_i}]\}_{i\in[d]}, \{\mathsf{KBF1}[\mathsf{P}^{(i)}_{\kappa_i}]\}_{i\in[d]})$ (we do not consider DS for a subset of $\mathcal{P}_\kappa$, because it can be embedded into $\mathsf{DS}[\mathcal{P}_\kappa]$). Let f be a deterministic procedure defined as $f(\mathcal{P}_\kappa) = \mathcal{P}_\kappa \cup \bar{\mathcal{P}}_\kappa$. Denote $f \circ \ldots \circ f(\mathcal{P}_\kappa)$ where f appears c times by $f_c(\mathcal{P}_\kappa)$. Then, we have the following theorem.

Theorem 8. *For all constant c and predicate family sets $\mathcal{P}_\kappa = (\mathsf{P}^{(1)}_{\kappa_1}, \ldots, \mathsf{P}^{(d)}_{\kappa_d})$, each of whose elements has a corresponding PES with (ζ, ℓ)-*KE-ind *for all constants $\zeta, \ell \in \mathbb{N}$, there exists a constant ζ' such that $\mathsf{P}_\kappa \in f_c(\mathcal{P}_\kappa)$ has a PES that satisfies (ζ', ζ')-*KE-ind *under the $\mathcal{D}_k$-MDDH assumption.*

Proof. Let $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_d)$ be PESs for $(\mathsf{P}^{(1)}_{\kappa_1}, \ldots, \mathsf{P}^{(d)}_{\kappa_d})$, respectively. We can construct a PES Γ for P by applying PES transformations in Sections 4.1 to 4.3 to $\mathbf{\Gamma}$ multiple times. Let δ be the maximum number of Dual-Trans that is applied to each single PES Γ_i to obtain Γ. For instance, δ in the following PES is 2 because the first Γ_2 is transformed by Dual-Trans twice, and the others are transformed by Dual-Trans less that twice.

$$\mathsf{KBF1\text{-}Trans}\left(\mathsf{DS\text{-}Trans}\left(\mathsf{Dual\text{-}Trans}\left(\mathsf{DS\text{-}Trans}\left(\Gamma_1, \mathsf{Dual\text{-}Trans}\left(\Gamma_2\right)\right)\right), \Gamma_2, \Gamma_3\right)\right).$$

Then, it is not hard to see that we can construct Γ with (ζ', ζ')-KE-ind for $\zeta' = \delta + 1$. This directly follows from Theorems 5 to 7. $\square$

Corollary 2. *Let $\mathcal{P}_\kappa = (\mathsf{P}^{(1)}_{\kappa_1}, \ldots, \mathsf{P}^{(d)}_{\kappa_d})$ be predicate families that have a PES with single-variable PMH. Then, we have a PES for $\mathsf{P}_\kappa \in f_c(\mathcal{P}_\kappa)$ with (ζ', ζ')-*KE-ind *for a constant ζ' under the $\mathcal{D}_k$-MDDH assumption, where $\zeta' - 1$ is the maximum number of* Dual *applied to each single predicate $\mathsf{P}^{(i)}_{\kappa_i}$ to obtain P_κ.*

This corollary directly follows from Theorems 4 and 8.

5 ABE from PES

In this section, we present our ABE scheme. We can construct an ABE scheme for any predicate family P_κ and a corresponding PES obtained in our framework if the PES satisfies (ζ, ζ)-KE-ind for some constant $\zeta \in \mathbb{N}$.

Construction. Let $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ be a PES with (ζ, ζ)-KE-ind for a predicate family $\mathsf{P}_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0, 1\}$. Then, we can construct an ABE scheme for predicate P_κ as follows.

$\mathsf{Setup}(1^\lambda, \kappa)$**:** Parse par from κ. It outputs pk and msk as follows.

$$\begin{aligned}
&\omega \leftarrow \mathsf{Param}(\mathsf{par}), \ \ \mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \ \ \overline{\mathbf{A}}, \overline{\mathbf{B}} \leftarrow \mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)}, \ \ \mathbf{h} \leftarrow \mathbb{Z}_p^{k+\zeta}, \\
&\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_\omega) \leftarrow (\mathbb{Z}_p^{(k+\zeta)\times(k+\zeta)})^\omega, \\
&\mathsf{pk}{=}(\mathbb{G}, [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_\omega^\top \mathbf{A}]_1, [\mathbf{A}^\top \mathbf{h}]_\mathsf{T}), \ \mathsf{msk}{=}(\mathbf{B}, \mathbf{h}, \mathbf{W}_1, \ldots, \mathbf{W}_\omega).
\end{aligned}$$

$\mathsf{Enc}(\mathsf{pk}, x, M)$**:** It takes pk, $x \in \mathcal{X}_\kappa$, and $M \in G_\mathsf{T}$ as inputs, and outputs ct_x by computing as follows.

$$(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x), \;\; \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k, \;\; \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$$
$$\mathbf{S} = (\mathbf{A}\mathbf{s}_0, \mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{n_1}), \;\; \widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2})$$
$$\mathsf{ct}_x = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3) = ([\mathbf{S}]_1, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_1, [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{h}]_\mathsf{T} M).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$**:** It takes pk, msk, and $y \in \mathcal{Y}_\kappa$ as inputs, and outputs sk_y by computing as follows.

$$(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})) \leftarrow \mathsf{EncKey}(y), \;\; \mathbf{r}_1, \ldots, \mathbf{r}_{m_1} \leftarrow \mathbb{Z}_p^k, \;\; \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2} \leftarrow \mathbb{Z}_p^{k+\zeta}$$
$$\mathbf{R} = (\mathbf{B}\mathbf{r}_1, \ldots, \mathbf{B}\mathbf{r}_{m_1}), \;\; \widehat{\mathbf{R}} = (\mathbf{h}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2})$$
$$\mathsf{sk}_y = (\mathsf{sk}_1, \mathsf{sk}_2) = ([\mathbf{R}]_2, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_2).$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_x, \mathsf{sk}_y)$**:** It takes pk, $\mathsf{ct}_x = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$, and $\mathsf{sk}_y = (\mathsf{sk}_1, \mathsf{sk}_2)$ such that $\mathsf{P}_\kappa(x, y) = 1$. Let $(\mathbf{E}, \overline{\mathbf{E}}) \leftarrow \mathsf{Pair}(x, y)$. It outputs $M' = \mathsf{ct}_3 / \Omega$ where

$$\Omega = \prod_{\substack{i \in [n_1+1] \\ j \in [m_3]}} e(\mathsf{ct}_{1,i}, \mathsf{sk}_{2,j})^{e_{i,j}} \cdot \prod_{\substack{i \in [n_3] \\ j \in [m_1]}} e(\mathsf{ct}_{2,i}, \mathsf{sk}_{1,j})^{\bar{e}_{i,j}}, \tag{4}$$

and where $\mathsf{ct}_{i,j}$ and $\mathsf{sk}_{i,j}$ refer to the j-th element of ct_i and sk_i, respectively, and $e_{i,j}$ and $\bar{e}_{i,j}$ refer to the (i, j)-th element of $\mathbf{E}$ and $\overline{\mathbf{E}}$, respectively.

Correctness. Let $\mathbf{c} = (c_1, \ldots, c_{n_3})$ and $\mathbf{k} = (k_1, \ldots, k_{m_3})$ be the outputs of $\mathsf{EncCt}(x)$ and $\mathsf{EncKey}(y)$, respectively, where

$$c_i = \sum_{z \in [n_2]} \theta_{i,z} \hat{s}_z + \sum_{t \in [n_1]^+, f \in [\omega]} \theta_{i,t,f} w_f s_t$$
$$k_j = \phi_j \alpha + \sum_{u \in [m_2]} \phi_{j,u} \hat{r}_u + \sum_{v \in [m_1], z \in [\omega]} \phi_{j,v,z} w_z r_v.$$

Then, we have

$$e(\mathsf{ct}_{1,i}, \mathsf{sk}_{2,j})^{e_{i,j}} = \left[e_{i,j} \mathbf{s}_{i-1}^\top \mathbf{A}^\top \left(\phi_j \mathbf{h} + \sum_{u \in [m_2]} \phi_{j,u} \hat{\mathbf{r}}_u + \sum_{v \in [m_1], z \in [\omega]} \phi_{j,v,z} \mathbf{W}_z \mathbf{B} \mathbf{r}_v \right) \right]_\mathsf{T}$$
$$e(\mathsf{ct}_{2,i}, \mathsf{sk}_{1,j})^{\bar{e}_{i,j}} = \left[\bar{e}_{i,j} \left(\sum_{z \in [n_2]} \theta_{i,z} \hat{\mathbf{s}}_z^\top + \sum_{t \in [n_1]^+, f \in [\omega]} \theta_{i,t,f} \mathbf{s}_t^\top \mathbf{A}^\top \mathbf{W}_f \right) \mathbf{B} \mathbf{r}_j \right]_\mathsf{T}.$$

From the correctness of the pair encoding scheme, we have

$$\sum_{\substack{i \in [n_1+1] \\ j \in [m_3]}} e_{i,j} s_{i-1} k_j + \sum_{\substack{i \in [n_3] \\ j \in [m_1]}} \bar{e}_{i,j} c_i r_j = \alpha s_0.$$

This corresponds to the following equation:

$$\prod_{\substack{i \in [n_1+1] \\ j \in [m_3]}} e(\mathsf{ct}_{1,i}, \mathsf{sk}_{2,j})^{e_{i,j}} \cdot \prod_{\substack{i \in [n_3] \\ j \in [m_1]}} e(\mathsf{ct}_{2,i}, \mathsf{sk}_{1,j})^{\bar{e}_{i,j}} = [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{h}]_\mathsf{T}.$$

Hence, we can see that $M = M'$.

Theorem 9. *Suppose Γ satisfies (ζ,ζ)-KE-ind. Then, our ABE scheme is adaptively secure under the $\mathcal{D}_k$-MDDH assumption. Let q_{sk} be the maximum number of $\mathcal{A}$'s queries to* KeyGen. *For any PPT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}}(\lambda) \le \mathsf{Adv}_{\mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + q_{\mathsf{sk}}\mathsf{Adv}_{\mathcal{B}_2,\Gamma}^{(\zeta,\zeta)\text{-KE-ind}}(\lambda).$$

Proof. The proof follows the dual system methodology [35]. We consider a series of hybrids H_1 and $\mathsf{H}_{2,j}$ for $j \in [q_{\mathsf{sk}}]$. To define each hybrid, we introduce a so-called semi-functional (SF) ciphertext and secret key, which are generated differently from normal ones. Specifically, an SF-ciphertext is generated as

$$\begin{aligned}
&(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x),\ \ \mathbf{s}_1, \ldots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k,\ \ \boxed{\mathbf{c}_0}, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta},\\
&\mathbf{S} = (\boxed{\mathbf{c}_0}, \mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{n_1}),\ \ \widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2}),\\
&\mathsf{ct}_x = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3) = ([\mathbf{S}]_1, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_1, [\boxed{\mathbf{c}_0^\top}\mathbf{h}]_{\mathsf{T}} M).
\end{aligned}$$

An SF-secret key is generated as

$$\begin{aligned}
&(m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{w})) \leftarrow \mathsf{EncKey}(y),\\
&\mathbf{r}_1, \ldots, \mathbf{r}_{m_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2} \leftarrow \mathbb{Z}_p^{k+\zeta},\ \ \boxed{\mu \leftarrow \mathbb{Z}_p},\\
&\mathbf{R} = (\mathbf{B}\mathbf{r}_1, \ldots, \mathbf{B}\mathbf{r}_{m_1}),\ \ \widehat{\mathbf{R}} = (\mathbf{h} + \boxed{\mu\mathbf{a}_\zeta^*}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2}),\\
&\mathsf{sk}_y = (\mathsf{sk}_1, \mathsf{sk}_2) = ([\mathbf{R}]_2, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_2).
\end{aligned} \tag{5}$$

In the hybrids, the distribution of secret keys and the challenge ciphertext are modified as follows:

H_1**:** Same as the original game G except that the challenge ciphertext is SF.
$\mathsf{H}_{2,j}(j \in [q_{\mathsf{sk}}])$**:** Same as H_1 except that the first j secret keys given to $\mathcal{A}$ are SF.

We prove that $\mathsf{G} \approx_c \mathsf{H}_1 \approx_c \mathsf{H}_{2,1} \approx_c, \ldots, \approx_c \mathsf{H}_{2,q_{\mathsf{sk}}}$ and $\mathcal{A}$'s advantage in $\mathsf{H}_{2,q_{\mathsf{sk}}}$ is statistically close to 0. We capture these as Lemmata 12 to 14. From these and the fact $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ABE}}(\lambda) = |\Pr[\langle\mathcal{A}, \mathsf{G}\rangle = \beta] - 1/2|$, we have that Theorem 9 holds. □

Lemma 12. *For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

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

Proof. To show this, we describe $\mathcal{B}$, which is given an instance of the $\mathcal{U}_{k+\zeta,k}$-MDDH problem $(\mathbb{G}, [\mathbf{A}]_1, [\mathbf{t}_\beta]_1)$. Note that we can write $\mathbf{t}_0 = \mathbf{A}\mathbf{s}_0$ and $\mathbf{t}_1 = \mathbf{c}_0$ where $\mathbf{s}_0 \leftarrow \mathbb{Z}_p^k$ and $\mathbf{c}_0 \leftarrow \mathbb{Z}_p^{k+\zeta}$.

1. $\mathcal{B}$ generates $\overline{\mathbf{B}}$, $\mathbf{h}$, and $\mathbf{W}_1, \ldots, \mathbf{W}_\omega$ by itself.
2. $\mathcal{B}$ computes $\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{W}_1^\top\mathbf{A}]_1, \ldots, [\mathbf{W}_\omega^\top\mathbf{A}]_1, [\mathbf{A}^\top\mathbf{h}]_{\mathsf{T}})$ and gives it to $\mathcal{A}$.
3. For queries $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes sk_y honestly. This is possible because $\mathcal{B}$ generates all elements in msk by itself.
4. For the challenge query with messages (M_0, M_1) and an attribute x^*, $\mathcal{B}$ flips the coin $\delta \leftarrow \{0,1\}$ and generates ct_{x^*} as

$$\begin{aligned}
&(n_1, n_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{w})) \leftarrow \mathsf{EncCt}(x),\ \ \mathbf{s}_1, \ldots, \mathbf{s}_{n_1} \leftarrow \mathbb{Z}_p^k,\ \ \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2} \leftarrow \mathbb{Z}_p^{k+\zeta}\\
&\mathbf{S} = (\mathbf{t}_\beta, \mathbf{A}\mathbf{s}_1, \ldots, \mathbf{A}\mathbf{s}_{n_1}),\ \ \widehat{\mathbf{S}} = (\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{n_2})\\
&\mathsf{ct}_x = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3) = ([\mathbf{S}]_1, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_1, [\mathbf{t}_\beta^\top\mathbf{h}]_{\mathsf{T}} M_\delta).
\end{aligned}$$

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

Clearly, the case $\beta = 0$ corresponds to G and the case $\beta = 1$ corresponds to H_1. □

Lemma 13. *Let $\mathsf{H}_{2,0} = \mathsf{H}_1$. For all PPT adversaries $\mathcal{A}$ and $j \in [q_{\mathsf{sk}}]$, there exists a PPT adversary $\mathcal{B}$ such that*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{2,j-1}\rangle = \beta] - \Pr[\langle \mathcal{A}, \mathsf{H}_{2,j}\rangle = \beta]| \leq \mathsf{Adv}^{(\zeta,\zeta)\text{-}\mathsf{KE}\text{-}\mathsf{ind}}_{\mathcal{B},\Gamma}(\lambda).$$

Proof. To show this, we describe $\mathcal{B}$, which is given an input of (ζ,ζ)-KE-ind game for $\eta = 1$, $(\mathbb{G}, [\mathbf{A}]_1, [\mathbf{B}]_2, \mathbf{a}^*_\zeta, \{[\mathbf{W}_i^\top \mathbf{A}]_1, [\mathbf{W}_i\mathbf{B}]_2\}_{i\in[\omega]})$.

1. $\mathcal{B}$ samples $\mathbf{h} \leftarrow \mathbb{Z}_p^{k+\zeta}$ and computes $[\mathbf{A}^\top \mathbf{h}]_{\mathsf{T}}$.
2. $\mathcal{B}$ gives $\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_\omega^\top \mathbf{A}]_1, [\mathbf{A}^\top \mathbf{h}]_{\mathsf{T}})$ to $\mathcal{A}$.
3. For the first $j-1$ queries $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes a SF key sk_y as shown in Eq. (5) using $\mathbf{h}$, $\mathbf{a}^*_\zeta$, $[\mathbf{B}]_2$, and $\{[\mathbf{W}_i\mathbf{B}]_2\}_{i\in[\omega]}$.
4. For the j-th query $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ queries $\mathcal{O}_\mathcal{Y}$ on y and $\mathbf{h}$, and obtains $([\mathbf{R}]_2, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_2)$, where the first element of $\widehat{\mathbf{R}}$ is $\mathbf{h} + \beta\mu\mathbf{a}^*_\zeta$. $\mathcal{B}$ returns $\mathsf{sk}_y = ([\mathbf{R}]_2, [\mathbf{k}(\mathbf{R}, \widehat{\mathbf{R}}, \mathbb{W})]_2)$ to $\mathcal{A}$.
5. For the rest of queries $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, y)$, $\mathcal{B}$ computes a normal secret key sk_y using $\mathbf{h}$, $[\mathbf{B}]_2$, and $\{[\mathbf{W}_i\mathbf{B}]_2\}_{i\in[\omega]}$.
6. For the challenge query with x^* and (M_0, M_1), $\mathcal{B}$ flip the coin $\delta \leftarrow \{0,1\}$ and queries $\mathcal{O}_\mathcal{X}$ on x^*. $\mathcal{B}$ obtains the reply $([\mathbf{S}]_1, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_1)$ and gives $\mathsf{ct}_{x^*} = ([\mathbf{S}]_1, [\mathbf{c}(\mathbf{S}, \widehat{\mathbf{S}}, \mathbb{W})]_1, [\mathbf{c}_0^\top \mathbf{h}]_{\mathsf{T}} M_\delta)$, where $\mathbf{c}_0$ is the first element of $\mathbf{S}$.
7. $\mathcal{B}$ outputs true$(\delta = \delta')$, where δ' is an output of $\mathcal{A}$.

Observe that the j-th secret key is identically distributed to the normal key if $\beta = 0$ and the SF-key otherwise. Thus, the case $\beta = 0$ corresponds to $\mathsf{H}_{2,j-1}$ and the case $\beta = 1$ corresponds to $\mathsf{H}_{2,j}$. □

Lemma 14. *For all PPT adversaries $\mathcal{A}$, we have*

$$|\Pr[\langle \mathcal{A}, \mathsf{H}_{2,q_{\mathsf{sk}}}\rangle = \beta] - 1/2| \leq 2^{-\Omega(\lambda)}.$$

Proof. Because $(\mathbf{A}^*||\mathbf{a}^*_1||\ldots||\mathbf{a}^*_\zeta)$ forms a basis of $\mathbb{Z}_p^{k+\zeta}$, redefining $\mathbf{h}$ as $\mathbf{h} = \mathbf{A}^*\mathbf{z} + \sum_{i\in[\zeta]} z_i \mathbf{a}^*_i$ where $\mathbf{z} \leftarrow \mathbb{Z}_p^k$, $z_i \leftarrow \mathbb{Z}_p$ does not change its distribution. Recall that the information on $\mathbf{h}$ that $\mathcal{A}$ obtains throughout the game is $\mathbf{A}^\top\mathbf{h}$ in pk, $\mathbf{h} + \mu\mathbf{a}^*_\zeta$ in sk_y, and $\mathbf{c}_0^\top\mathbf{h}$ in ct_{x^*}. $\mathbf{A}^\top\mathbf{h}$ does not contain the information on z_ζ because $\mathbf{A}^\top\mathbf{a}^*_\zeta = \mathbf{0}$. Similarly, each $\mathbf{h} + \mu\mathbf{a}^*_\zeta$ in secret keys also does not contain the information on z_ζ because it is masked by the fresh randomness μ. Thus, $\mathbf{c}_0^\top\mathbf{h} = \mathbf{c}_0^\top(\mathbf{A}^*\mathbf{z} + \sum_{i\in[\zeta]} z_i\mathbf{a}^*_i)$ is randomly distributed in $\mathbb{Z}_p$ for $\mathcal{A}$ unless $\mathbf{c}_0^\top\mathbf{a}^*_\zeta = 0$ because z_ζ is randomly distributed for $\mathcal{A}$. Since $\mathbf{c}_0$ is randomly chosen from $\mathbb{Z}_p^{k+\zeta}$, $\mathbf{c}_0^\top\mathbf{a}^*_\zeta = 0$ with a probability $2^{-\Omega(\lambda)}$. If it is not the case, ct_{x^*} does not have information on β because ct_3 is randomly distributed in G_{T}. □

6 Extensions, Instantiations, and Applications

In this section, we provide extensions, instantiations, and applications of our framework. We first provide an overview.

6.1 Overview

We obtain many applications in an analogous manner to the applications in [8].

Extended Framework. On the framework level, we obtain key-policy augmentation over *a set of predicate families*, denoted KBF, which is more powerful than the augmentation over a *single* predicate family ($\mathsf{KBF1}$), as done in §4.3. This follows exactly the same modular approach as in [8]. That is, in our context, we can show that KBF is implied by $\mathsf{KBF1}$ together with the direct sum and $\mathsf{CBF1}_{\mathsf{OR}}$. We explain this in §6.2. Moreover, more applications such as nested-policy ABE can also be obtained analogously to [8].

New Instantiations. On the instantiation level, we have showed the result overview in the introduction. Here, we briefly describe how to obtain such instantiations. The full details follow from §6.3.

– Completely unbounded ABE for monotone Boolean formulae. Analogously to [8], we have that this predicate (in the key-policy flavor) is exactly $\mathsf{KBF1}[\mathsf{P}^{\mathsf{IBBE}}]$, where $\mathsf{P}^{\mathsf{IBBE}}$ is the predicate for ID-based broadcast encryption. IBBE can then be augmented from IBE, of which we know a PMH-secure PES from *e.g.,* [6]. The CP flavor is obtained by the dual conversion.
– Completely unbounded ABE for non-monotone Boolean formulae (the OSW type). This is also analogous to [8], where we consider two-mode IBBE (TIBBE), which can be then obtained by IBE and its negated predicate.
– Non-monotone KP-ABE with constant-size ciphertexts. A monotone variant is obtained by simply using the PMH-secure PES for IBBE with constant-size ciphertext encodings. Such a PES can be extracted from the PES for doubly spatial predicate in [6]. Since our $\mathsf{KBF1\text{-}Trans}$ preserves ciphertext encoding sizes, the converted scheme also obtains constant-size ciphertext encodings. For the non-monotone case, such a PES for TIBBE can be obtained by the disjunction of IBBE and negated IBBE (NIBBE). The latter can be viewed as a special case of negated doubly spatial predicate in [6], of which PES with constant-size encodings was reported. In §6.7, we directly construct a new TIBBE, which is two times efficient than the generic one from the disjunction.
– CP-ABE with constant-size ciphertexts. First note that we consider schemes with some bound on the size of policies (Boolean formulae), which the same requirement as CP-ABE with constant-size ciphertexts of [1, 8, 9]. We obtain this by two steps. First we show that, when considering small-universe, KP-ABE implies CP-ABE (for Boolean formulae, with the bounded condition). We use the depth-universal circuit [17] in this conversion. Second we show that CP-ABE with small universe implies CP-ABE with large universe (again for Boolean formulae, with the bounded condition). To the best of our knowledge, these conversions were not known and can be of an independent interest, as they are applied to ABE in general (not necessarily to PES). Note that we cannot do that as Attrapadung *et al.* [9] did, who considered similar implications in the case of more powerful *span programs*.
– ABE with constant-size keys. CP/KP-ABE with constant-size keys is obtained by the dual of KP/CP-ABE with constant-size ciphertexts, respectively.

As examples, we provide the descriptions of three concrete instantiations in §A.

New Applications. As a new application, we provide a new unified predicate related to *non-monotone* ABE. Previously, there are two types of non-monotone ABE: the OSW type (Ostrovsky, Sahai, and Waters [30]) and the OT type (Okamoto and Takashima [29]). In the OSW type, a sub-predicate $P(y, X)$ amounts to check if an attribute is not in a set, *e.g.,* if $y \notin X$, while the OT type, a label tag is also attached, but a sub-predicate $P'((\text{tag}, y), (\overline{\text{tag}}, x))$ only checks the inequality on the same tag, *e.g.,* if $\text{tag} = \overline{\text{tag}} \wedge y \neq x$. Intuitively, the OSW type has a disadvantage in that the non-membership test takes the complement over the *whole universe* and this may be too much for some applications, where we would like to consider multiple sub-universe and confine the complement to only in the related sub-universe. On the other hand, the OT type confines the non-membership to those with the same tag, but the non-membership test is enabled only with the set of single element, *e.g.,* $\{x\}$. We unify both types to overcome both disadvantages; that is, a sub-predicate $P'((\text{tag}, y), (\overline{\text{tag}}, X))$ would check if $\text{tag} = \overline{\text{tag}} \wedge y \notin X$. We remark that when considering large-universe *monotone* ABE, there is no benefit to consider multiple spaces, since $\mathbb{Z}_p$ is already exponentially large, and we can just treat a hashed value $H(\text{tag}, y)$ as an attribute in $\mathbb{Z}_p$. In non-monotone ABE, we have to check the equality (of tags) and the non-membership at once, and the approach by hashing does not work. We motivate more on the unified non-monotone ABE, and provide definitions and constructions in §6.4.

6.2 Augmentation over Predicate Sets

Following the composition framework of [8], we can also analogously define key-policy Boolean formula augmentations over a *set* of predicate families, rather than only a *single* predicate family, as done in Definition 14.

Some Terminology. Throughout this subsection, let $\mathcal{P} = \{\mathsf{P}^{(1)}, \ldots, \mathsf{P}^{(k)}\}$ be a set of predicate families. Each family $\mathsf{P}^{(j)} = \{P^{(j)}_{\kappa_j}\}_{\kappa_j}$ is indexed by $\kappa_j = (N, \mathsf{par}_j)$. The domain for each predicate is specified by $\mathsf{P}^{(j)}_{\kappa_j} : \mathcal{X}^{(j)}_{\kappa_j} \times \mathcal{Y}^{(j)}_{\kappa_j} \to \{0,1\}$. Unless specified otherwise, we define the combined index as $\kappa = (N, \mathsf{par}) = (N, (\mathsf{par}_1, \ldots, \mathsf{par}_k))$. Let $\mathbb{X}_\kappa \coloneqq \bigcup_{i\in[k]}(\{i\} \times \mathcal{X}^{(i)}_{\kappa_i})$ and $\mathbb{Y}_\kappa \coloneqq \bigcup_{i\in[k]}(\{i\} \times \mathcal{Y}^{(i)}_{\kappa_i})$.

Definition 16 (Key-Policy Augmentation over Predicate Sets). Let $\mathcal{P} = \{\mathsf{P}^{(1)}, \ldots, \mathsf{P}^{(k)}\}$ be a set of predicate families. We define the predicate for *key-policy Boolean formula augmentation over set* $\mathcal{P}$ as $\mathsf{KBF}[\mathcal{P}] = \{\overline{\mathsf{P}}_\kappa\}_\kappa$ where $\overline{\mathsf{P}}_\kappa : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{0,1\}$ by letting

- $\bar{\mathcal{X}}_\kappa = 2^{\mathbb{X}_\kappa}$.
- $\bar{\mathcal{Y}}_\kappa = \bigcup_{i\in\mathbb{N}}(\mathbb{Y}^i_\kappa \times \mathcal{F}_i)$, where $\mathcal{F}_i$ consists of all monotone Boolean formulae with input length i.
- For $X \in \bar{\mathcal{X}}_\kappa$ and $Y = (((j_1, y_1), \ldots, (j_n, y_n)), f) \in \bar{\mathcal{Y}}_\kappa$ where $f : \{0,1\}^n \to \{0,1\}$, we define $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ by setting

$$b_i = 1 \ \text{ iff } \ \exists (j_i, x) \in X \text{ s.t. } \mathsf{P}^{(j_i)}_{\kappa_{j_i}}(x, y_i) = 1$$

  We then define $\overline{\mathsf{P}}_\kappa(X, y) = 1 \Leftrightarrow f(b) = 1$.

Unbounded/Dynamic/Static. Similarly to [8], we consider (confined) variants of the predicate $\mathsf{KBF}[\mathcal{P}]$ as follows. Consider the domain of $((j_1, \ldots, j_n), f)$, which specifies a policy over predicates. We denote their full domain as D, which can be inferred from Definition 16. For a class $C \subseteq D$, the predicate $\mathsf{KBF}[\mathcal{P}]$ with the domain of $((j_1, \ldots, j_n), f)$ being confined to C is denoted by $\mathsf{KBF}_C[\mathcal{P}]$ and is also called *dynamic Boolean formula composition with class* C. It is called *unbounded* if $C = D$. It is called *static* if $|C| = 1$. We will use a terminology $\mathsf{P}^{(1)} \vee \mathsf{P}^{(2)}$ and $\mathsf{P}^{(1)} \wedge \mathsf{P}^{(2)}$ to naturally denote the static OR and AND composition over $\{\mathsf{P}^{(1)}, \mathsf{P}^{(2)}\}$, respectively.

We have the following lemma, which follows analogously to the case of span programs in [8].

Lemma 15. $\mathsf{KBF}[\mathcal{P}]$ *can be embedded into* $\mathsf{KBF1}[\mathsf{CBF1}_{\mathsf{OR}}[\mathsf{DS}[\mathcal{P}]]]$.

6.3 Basic Predicates

In the following subsections, we describe a modular approach similarly to [8] in obtaining our ABE instantiations from simpler basic predicates. We first recapitulate the following basic predicates as follows. For abbreviations, IBE is for ID-based encryption; NIBE is for negated IBE, IBBE is for ID-based broadcast encryption [18]; IBR is for ID-based revocation [10]; and TIBBE is for two-mode IBBE [38]. Let the universe in these predicates be $\mathcal{U} = \mathbb{Z}_p$.

- $\mathsf{P}^{\mathsf{IBE}} : \mathcal{U} \times \mathcal{U} \to \{0,1\}$ defined as $\mathsf{P}^{\mathsf{IBE}}(x, y) = 1 \Leftrightarrow x = y$.
- $\mathsf{P}^{\mathsf{NIBE}} : \mathcal{U} \times \mathcal{U} \to \{0,1\}$ defined as $\mathsf{P}^{\mathsf{NIBE}}(x, y) = 1 \Leftrightarrow x \neq y$.
- $\mathsf{P}^{\mathsf{IBBE}} : 2^{\mathcal{U}} \times \mathcal{U} \to \{0,1\}$ defined as $\mathsf{P}^{\mathsf{IBBE}}(X, y) = 1 \Leftrightarrow y \in X$.
- $\mathsf{P}^{\mathsf{IBR}} : 2^{\mathcal{U}} \times \mathcal{U} \to \{0,1\}$ defined as $\mathsf{P}^{\mathsf{IBR}}(X, y) = 1 \Leftrightarrow y \notin X$.
- $\mathsf{P}^{\mathsf{TIBBE}} : 2^{\mathcal{U}} \times (\{1,2\} \times \mathcal{U}) \to \{0,1\}$ defined as

$$\mathsf{P}^{\mathsf{TIBBE}}(X, (i, y)) = 1 \Leftrightarrow (i = 1 \wedge y \in X) \vee (i = 2 \wedge y \notin X).$$

It is straightforward to see that $\mathsf{P}^{\mathsf{IBBE}}$ can be embedded into $\mathsf{CBF1}_{\mathsf{OR}}[\mathsf{P}^{\mathsf{IBE}}]$, while $\mathsf{P}^{\mathsf{IBR}}$ can be embedded into $\mathsf{CBF1}_{\mathsf{AND}}[\mathsf{P}^{\mathsf{NIBE}}]$, and $\mathsf{P}^{\mathsf{TIBBE}}$ can be embedded into $\mathsf{CBF1}_{\mathsf{OR}}[\mathsf{P}^{\mathsf{IBBE}} \odot \mathsf{P}^{\mathsf{IBR}}]$. Now since we have a PES instantiation for $\mathsf{P}^{\mathsf{IBE}}$ and $\mathsf{P}^{\mathsf{NIBE}}$ that is secure in the sense of perfect master-key hiding from [6] and [5], respectively, we can instantiate ABE for these predicates via our transformations.

6.4 Completely Unbounded ABE for Monotone Formulae

We denote by $\mathsf{P}^{\mathsf{KP\text{-}MBF}}$ the predicate of key-policy ABE for monotone boolean formulae (MBF) where all the parameters (the policy size, the attribute set size, the number of allowed multi-use of attributes in one policy) are unbounded and the attribute universe $\mathcal{U}$ is super-polynomially large. Its precise definition can be obtained modularly as

$$\mathsf{P}^{\mathsf{KP\text{-}MBF}} := \mathsf{KBF1}[\mathsf{P}^{\mathsf{IBBE}}],$$

or equivalently, it is $\mathsf{KBF}[\mathsf{P}^{\mathsf{IBE}}]$. The ciphertext-policy flavor is its dual, namely, $\mathsf{P}^{\mathsf{CP\text{-}MBF}} := \mathsf{Dual}[\mathsf{P}^{\mathsf{KP\text{-}MBF}}]$.

Kowalczyk and Wee [25] recently proposed such an unbounded KP-ABE under the MDDH assumption; however, the ciphertext-policy variant has remained as an open problem. By using our modular transformation KBF1-Trans and Dual-Trans to the PES for $\mathsf{P}^{\mathsf{IBBE}}$ (which is obtained via transformations to the IBE of [6], respectively), we obtain the first such unbounded CP-ABE under the MDDH assumption.

6.5 Completely Unbounded ABE for Non-Monotone Formulae

Due to De Morgan's Law, any non-monotone boolean formula (NBF) can be expressed by another formula where all the NOT gates are applied only at the input values. Using this fact, we can define the predicate for the completely-unbounded (key-policy) ABE for non-monotone Boolean formulae, denoted by $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}}$, as

$$\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}} := \mathsf{KBF1}[\mathsf{P}^{\mathsf{TIBBE}}].$$

This type of ABE for NBF was defined by Ostrovsky, Sahai and Waters [30], and hence we call it the OSW-type. It is crucial to note that, when we consider large-universe schemes, ABE for NBF is not trivially implied from ABE for MBF. One trivial implementation (that does not work) prepares the negative version of all attributes in the universe and requires any attribute set, say S, to include all negative attributes, say $\bar{x}$, if x is not in S; however, this is not possible due to the super-polynomial size universe.

The Okamoto-Takashima type [29] of ABE for NBF was defined differently. Its modular definition was captured in [8], and we recap it here. Let $\mathcal{L}$ be the "label" universe and $\mathcal{U}$ be the attribute universe. First define

$$\mathcal{X}^{\mathsf{OT}} = \{\, \{(a_1, x_1), \ldots, (a_t, x_t)\} \,|\, a_i \in \mathcal{L}, x_i \in \mathcal{U}, t \in \mathbb{N}, \text{if } i \neq j \text{ then } a_i \neq a_j \,\}.$$

We then define $\mathsf{P}^{\mathsf{OT}} : \mathcal{X}^{\mathsf{OT}} \times (\{1,2\} \times \mathcal{L} \times \mathcal{U}) \to \{0,1\}$ by

$$\begin{aligned}\mathsf{P}^{\mathsf{OT}}\big(\, \{(a_1, x_1), \ldots, (a_t, x_t)\},\ (i, \ell, y)\,\big) = 1 \ \Leftrightarrow\ & \big(i = 1 \wedge (\exists j : a_j = \ell \wedge x_j = y)\big) \vee \\ & \big(i = 2 \wedge (\exists j : a_j = \ell \wedge x_j \neq y)\big).\end{aligned}$$

The OT-type ABE for NBF can then be defined as

$$\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}} := \mathsf{KBF1}[\mathsf{P}^{\mathsf{OT}}].$$

Disadvantages of the Previous Two Types of Non-monotonicity. Intuitively, we can consider that there is one large space $\mathcal{U}$ as a ciphertext-attribute universe in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}}$, whereas there are multiple spaces $\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(t)}$ in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}}$. When we consider monotone ABE, there is no benefit to consider multiple spaces because $\mathcal{U}$ is already exponentially large, which concequently yields large universe ABE. However, the situation is different when we consider non-monotone ABE. That is, the negation in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}}$ is for the entire attribute universe $\mathcal{U}$, whereas that in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}}$ is for only a fraction of attribute universe, i.e., $\mathcal{U}^{(i)}$.

This is a critical difference in practice as pointed out by Tomida et al. [34]. Considering an example is the best way to describe the difference. Suppose that an attribute consists of a label and value, like YEAR:1991-2000, where YEAR is a label and 1991-2000 is a value. This is quite natural because each record in a typical relational database has this structure. Then, we consider the case where we handle two labels YEAR and CATEGORY. The negation in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}}$ (OSW-negation) can be described as (NOT YEAR:1991-2000) while negation in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}}$ (OT-negation) is like (YEAR:NOT 1991-2000). Semantically, the former implies that the policy is satisfied if attribute YEAR:1991-2000 does not exist in a attribute set. On the other hand, the latter implies that the policy is satisfied if an attribute set has an attribute on label YEAR and its attribute is not 1991-2000.

This semantical difference arises from the structural difference of attribute universes in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}}$ and $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}}$. In $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW}}$, one needs to embed the information on both label and value into $\mathcal{U}$. On the other hand, in $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}}$, one can associate the label with an index $i \in \mathcal{L}$ of the attribute universe and embed only the information on a value into $\mathcal{U}^{(i)}$.

For typical applications of ABE, the structure of the universe in OT-nagation is more desirable. Consider the case to increase labels in ABE system that is in operation. If the system is based on OSW-negation, some inconvenience arises. That is, a secret key whose policy is negation of an attribute whose label is a new one that the system has not supported before can decrypt all ciphertexts generated so far. Let one of the new labels be ARTIST. 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.

Nevertheless, OT-non-monotonicity is not still almighty. If we carefully look at the definition of $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OT}}$, we can see that each attribute sets can have at most one value for each label. That is, it does not allow attributes such as CATEGORY:Rock, Blues, R&B. This is also inconvenient when we consider labels that naturally takes multiple values per record or instance. This inconvenience motivate us to consider the following new type of non-monotone ABE, which does not cause the above problems.

New Unified Type of ABE for Non-Monotone Boolean Formulae. We propose a new "hybrid" type that combines and unifies both types (OSW,OT) above. First define

$$\mathcal{X}^{\mathsf{OSWOT}} = \{\, \{(a_1, X_1), \ldots, (a_t, X_t)\} \,|\, a_i \in \mathcal{L}, X_i \subseteq \mathcal{U}, t \in \mathbb{N}, \text{if } i \neq j \text{ then } a_i \neq a_j \,\}.$$

We then define $\mathsf{P}^{\mathsf{OSWOT}} : \mathcal{X}^{\mathsf{OT}} \times (\{1,2\} \times \mathcal{L} \times \mathcal{U}) \to \{0,1\}$ by

$$\begin{aligned}\mathsf{P}^{\mathsf{OSWOT}}\big(\, \{(a_1, X_1), \ldots, (a_t, X_t)\},\ (i, \ell, y)\,\big) = 1 \ \Leftrightarrow\ & \big(i = 1 \wedge (\exists j : a_j = \ell \wedge y \in X_j)\big) \vee \\ & \big(i = 2 \wedge (\exists j : a_j = \ell \wedge y \notin X_j)\big).\end{aligned}$$

The unified type ABE for NBF can then be defined as

$$\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSWOT}} := \mathsf{KBF1}[\mathsf{P}^{\mathsf{OSWOT}}].$$

We can instantiate ABE for NBF (in all types). For this purpose, it is sufficient to instantiate PES for $\mathsf{P}^{\mathsf{OSWOT}}$. Now, using the idea similar to [8], it is not difficult to see that the above $\mathsf{P}^{\mathsf{OSWOT}}$ can be embedded into

$$\mathsf{CBF1}_{\mathsf{OR}}[\mathsf{CBF1}_{\mathsf{OR}}[\mathsf{P}^{\mathsf{IBE}} \wedge \mathsf{P}^{\mathsf{IBBE}}] \odot \mathsf{CBF1}_{\mathsf{OR}}[\mathsf{P}^{\mathsf{IBE}} \wedge \mathsf{P}^{\mathsf{IBR}}]].$$

6.6 Unified Definition for Bounded ABE for Boolean Formulae

Towards constructing ABE with constant-size ciphertexts or keys, we will set bounds on some parameters. In this subsection, we give the following unified definition that can deal with combinations of bounds.

Definition 17 (Predicate variants of KP-ABE for MBF). The predicate family of KP-ABE for MBF in the xx variant, denoted by $\mathsf{P}_{\kappa}^{\text{KP-MBF-xx}} : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$, is define as follows. Each variant is indexed by a sub-vector of $\kappa = (U, T, N, M, D, \varphi) \in \mathbb{N}^6$ (see more below). Denote the attribute universe as $\mathcal{U}$ and let $U = |\mathcal{U}|$.

– $\mathcal{X}_\kappa \coloneqq \binom{\mathcal{U}}{\leq T} = \{X \in 2^{\mathcal{U}} : |X| \leq T\}$.
– $\mathcal{Y}_\kappa \coloneqq \bigcup_{n \leq N} (\mathcal{V}_{n,\varphi} \times \mathcal{F}_{n,M,D})$, where
  • $\mathcal{V}_{n,\varphi} \coloneqq \{\mathbf{y} \in \mathcal{U}^n \mid \text{The same element can appear at most } \varphi \text{ times in } \mathbf{y}.\}$.
  • $\mathcal{F}_{n,M,D}$ consists of all monotone Boolean formulae with input length n, and size at most M, depth at most D.
– For $X \in \mathcal{X}_\kappa$ and $Y = ((y_1, \ldots, y_n, f) \in \mathcal{Y}_\kappa$ where $f : \{0,1\}^n \to \{0,1\}$, we set $b_j = 1$ iff $y_j \in X$. Define $\mathsf{P}_{\kappa}^{\text{KP-MBF-xx}}(X, Y) = 1 \Leftrightarrow f(b_1, \ldots, b_n) = 1$.

The index κ shows bounds regarding the domains $\mathcal{X}_\kappa, \mathcal{Y}_\kappa$. For a predicate variant where some parameters are unbounded, we write - in κ; for example, a predicate with no bound T will be indexed by $(U, \text{-}, N, M, D, \varphi)$. Note that if U is unbounded, we set $\mathcal{U}$ as a super-polynomial-size space, in particular, $\mathcal{U} = \mathbb{Z}_p$ (such a variant is called "large-universe"). In this way, we can define a variant by the combination of the bounds, hence obtain up to 64 variants (some might be subsumed by others). We use $\mathsf{xx} \in \{0, \ldots, 63\}$ to name each variant by using the position of 1 in $(\mathsf{xx})_2$ to mean that there is a bound in the corresponding position in κ. We will particularly consider the following variants.

– $\mathsf{P}^{\text{KP-MBF-0}}$: the completely-unbounded predicate (*i.e.,* $\mathsf{P}^{\text{KP-MBF}}$).
– $\mathsf{P}^{\text{KP-MBF-16}}$, indexed by $\kappa = (\text{-}, T, \text{-}, \text{-}, \text{-}, \text{-})$, is a predicate with the bounded attribute set size t.
– $\mathsf{P}^{\text{KP-MBF-31}}$, indexed by $\kappa = (\text{-}, T, N, M, D, \varphi)$, is a predicate with large universe.
– $\mathsf{P}^{\text{KP-MBF-63}}$: the completely-bounded predicate.

We can also analogously define the variants for NBF, $\mathsf{P}^{\text{KP-NBF-T-xx}}$, where $\mathsf{T} \in \{\mathsf{OSW}, \mathsf{OT}, \mathsf{OSWOT}\}$ in a natural manner (details are omitted).

6.7 KP-ABE with Constant-Size Ciphertexts

Monotone KP-ABE with Constant-Size Ciphertexts. We consider the bounded-attribute-set-size predicate, more precisely, $\mathsf{P}^{\text{KP-MBF-16}}$. It can be interpreted as $\mathsf{KBF1}[\mathsf{P}^{\mathsf{IBBE}'}]$, where we define the predicate family $\mathsf{P}^{\mathsf{IBBE}'} = \{\mathsf{P}_T^{\mathsf{IBBE}'}\}$ indexed by $T \in \mathbb{N}$ by confining the domain $2^{\mathcal{U}}$ in $\mathsf{P}^{\mathsf{IBBE}'}$ to $\binom{\mathcal{U}}{\leq T}$.

Since our $\mathsf{KBF1\text{-}Trans}$ preserves the ciphertext encoding size, to obtain a PES for $\mathsf{P}^{\text{KP-MBF-16}}$ with constant-size ciphertext encodings, it suffices to construct such a PES for $\mathsf{P}^{\mathsf{IBBE}'}$. For a set $X \subsetneq \mathbb{Z}_p$, write

$$p_X(z) = \prod_{i \in X} (z - i) = a_0 + a_1 z + \cdots + a_T z^T$$

and define $\mathbf{v}_X \coloneqq (a_0, \ldots, a_T)^\top \in \mathbb{Z}_p^{T+1}$ and $\mathbf{v}'_X = (1, a_0, \ldots, a_T)^\top \in \mathbb{Z}_p^{T+2}$. For an element $y \in \mathbb{Z}_p$, define $\mathbf{M}_y \in \mathbb{Z}_p^{(T+1)\times T}$, $\mathbf{M}'_y \in \mathbb{Z}_p^{(T+2)\times(T+1)}$ as

$$\mathbf{M}_y \coloneqq \begin{pmatrix} \mathbf{m}_y^\top \\ \mathbf{I}_T \end{pmatrix} \coloneqq \begin{pmatrix} -y & -y^2 & \cdots & -y^T \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}, \qquad \mathbf{M}'_y \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{M}_y \end{pmatrix}.$$

It can be shown that $y \in X$ iff $p_X(y) = 0$ iff $\mathbf{v}_X$ is in the column span of $\mathbf{M}_y$. A PES for $\mathsf{P}^{\mathsf{IBBE}'}$ is constructed as follows.[4]

[4] Note that this PES can be viewed as a special case extracted from the PES for doubly spatial encryption predicate in [6].

– $\mathsf{Param}(T) = T + 2 = |\mathbf{w}|$.
– $\mathsf{EncCt}(X) = (0, 0, c)$ where $c = s\mathbf{w}^\top \mathbf{v}'_X$, where the non-lone variable is s.
– $\mathsf{EncKey}(y) = (1, 0, \mathbf{k})$ where $\mathbf{k} = (\alpha, \mathbf{0}) + r\mathbf{w}^\top \mathbf{M}'_y$, where the non-lone variable is r.
– $\mathsf{Pair}(X, y) = (\mathbf{e}, \bar{\mathbf{e}})$ where $\mathbf{e} = -(1, a_1, \ldots, a_T)^\top$ and $\bar{\mathbf{e}} = 1$.

The correctness amounts to prove $s\mathbf{e}^\top \mathbf{k}^\top + \mathbf{c}\bar{\mathbf{e}}\, r = \alpha s$, which can be shown as follows. From $y \in X$ we have $p_X(y) = 0$ and hence, by inspection, we have $\mathbf{M}'_y \mathbf{e} = -(1, a_0, \ldots, a_T)^\top = -\mathbf{v}'_X$, and this leads to the claim.

Lemma 16. *The above PES for* $\mathsf{P}^{\mathsf{IBBE}'}$ *satisfies perfect master-key hiding.*

Proof. Suppose $\mathsf{P}^{\mathsf{IBBE}'}(X, y) = 0$, *i.e.*, $y \notin X$. The encoding construction implies a system of equations with unknown $\alpha, \mathbf{w}$:

$$\begin{pmatrix} 1 & r & 0 \\ 0 & 0 & r\mathbf{M}_y^\top \\ 0 & s & s\mathbf{v}_X^\top \end{pmatrix} \begin{pmatrix} \alpha \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{k}^\top \\ \mathbf{c}^\top \end{pmatrix}$$

From $y \notin X$, we have $\mathbf{v}_X \notin \mathsf{span}(\mathbf{M}_y)$. Hence $(1, 0, \ldots, 0)$ is not in the row span of the matrix on the left-hand side. Therefore, α is completely hidden. □

Non-Monotone KP-ABE with Constant-Size Ciphertexts. Here, we consider the predicate $\mathsf{P}^{\mathsf{KP\text{-}NBF\text{-}OSW\text{-}16}}$. Similarly as above, it is equivalent to $\mathsf{KBF1}[\mathsf{P}^{\mathsf{TIBBE}'}]$, where $\mathsf{P}^{\mathsf{TIBBE}'}$ is defined analogously (confining to the attribute sets of size $\leq T$). We define $\mathbf{V}_X \in \mathbb{Z}_p^{(T+3)\times 2}$, $\mathbf{M}_y^{(1)} \in \mathbb{Z}_p^{(T+3)\times(T+1)}$, $\mathbf{M}_y^{(2)} \in \mathbb{Z}_p^{(T+3)\times(T+2)}$ as

$$\mathbf{V}_X \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{v}_X \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{M}_y^{(1)} \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{M}_y \\ 1 & 0 \end{pmatrix}, \qquad \mathbf{M}_y^{(2)} \coloneqq \begin{pmatrix} 1 & 0 & 0 \\ 1 & \mathbf{m}_y^\top & 0 \\ 0 & \mathbf{I}_T & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

A PES for $\mathsf{P}^{\mathsf{TIBBE}'}$ is constructed as follows.

– $\mathsf{Param}(T) = T + 3 = |\mathbf{w}|$.
– $\mathsf{EncCt}(X) = (0, 0, \mathbf{c})$ where $\mathbf{c} = s\mathbf{w}^\top \mathbf{V}_X$, where the non-lone variable is s.
– $\mathsf{EncKey}(i, y) = (1, 0, \mathbf{k})$ where $\mathbf{k} = (\alpha, \mathbf{0}) + r\mathbf{w}^\top \mathbf{M}_y^{(i)}$, where the non-lone variable is r.
– $\mathsf{Pair}(X, (i, y)) = (\mathbf{e}, \bar{\mathbf{e}})$, where we recall that we have the two following cases when $\mathsf{P}^{\mathsf{TIBBE}'}(X, (i, y)) = 1$.
  - If $i = 1$ and $y \in X$, we set $\mathbf{e} = (1, a_1, \ldots, a_T)^\top$ and $\bar{\mathbf{e}} = (-1, -1)^\top$.
  - If $i = 2$ and $y \notin X$, we set $\mathbf{e} = (1, \frac{a_1}{\delta}, \ldots, \frac{a_T}{\delta}, \frac{1}{\delta})^\top$ and $\bar{\mathbf{e}} = (-1, -\frac{1}{\delta})^\top$, where $\delta \coloneqq p_X(y) \neq 0$.

The correctness amounts to prove $s\mathbf{e}^\top \mathbf{k}^\top + \mathbf{c}\bar{\mathbf{e}}\, r = \alpha s$, which can be shown as follows.

– Suppose $i = 1$ and $y \in X$. Then, the above holds since

$$\mathbf{M}_y^{(1)} \mathbf{e} = (1, a_0, \ldots, a_T, 1)^\top \qquad \mathbf{V}_X \bar{\mathbf{e}} = -(1, a_0, \ldots, a_T, 1)^\top,$$

which implies $\mathbf{M}_y^{(1)} \mathbf{e} + \mathbf{V}_X \bar{\mathbf{e}} = 0$, and hence the claim.
– Suppose $i = 2$ and $y \notin X$. Then, the above holds since

$$\mathbf{M}_y^{(2)} \mathbf{e} = (1, 1 - \frac{a_1 y + \cdots a_T y^t}{\delta}, \frac{a_1}{\delta}, \ldots, \frac{a_T}{\delta}, \frac{1}{\delta})^\top$$
$$\mathbf{V}_X \bar{\mathbf{e}} = (-1, -\frac{a_0}{\delta}, \ldots, -\frac{a_T}{\delta}, -\frac{1}{\delta})^\top,$$

which implies $\mathbf{M}_y^{(2)} \mathbf{e} + \mathbf{V}_X \bar{\mathbf{e}} = 0$, and hence the claim.

Lemma 17. *The above PES for* $\mathsf{P}^{\mathit{TIBBE}'}$ *satisfies perfect master-key hiding.*

Proof. Suppose $\mathsf{P}^{\mathsf{TIBBE}'}(X,(i,y))=0$. We have two cases.

– Case $i=1$ and $y \notin X$. The encoding construction implies a system of equations with unknown $\alpha, \mathbf{w}$:

$$\begin{pmatrix} 1 & r & 0 & r \\ 0 & 0 & r\mathbf{M}_y^\top & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s\mathbf{v}_X^\top & s \end{pmatrix} \begin{pmatrix} \alpha \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{k}^\top \\ \mathbf{c}^\top \end{pmatrix}$$

From $y \notin X$, we have $\mathbf{v}_X \notin \mathsf{span}(\mathbf{M}_y)$. Hence $(1,0,\ldots,0)$ is not in the row span of the matrix on the left-hand side. Therefore, α is completely hidden.

– Case $i=2$ and $y \in X$. The encoding construction implies a system of equations with unknown $\alpha, \mathbf{w}$:

$$\begin{pmatrix} 1 & r & r & 0 & 0 \\ 0 & 0 & r\mathbf{m}_y & r\mathbf{I}_t & 0 \\ 0 & 0 & 0 & 0 & r \\ 0 & s & 0 & 0 & 0 \\ 0 & 0 & s\mathbf{v}_{X,0}^\top & s\mathbf{v}_{X,1\to T}^\top & s \end{pmatrix} \begin{pmatrix} \alpha \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{k}^\top \\ \mathbf{c}^\top \end{pmatrix},$$

where we write $\mathbf{v}_X^\top = \mathbf{v}_{X,0}^\top || \mathbf{v}_{X,1\to T}^\top = a_0 || (a_1 \ldots, a_T)$. By inspection, we have that $(r,0,\ldots,0)$ is not in the row span of $(r\mathbf{m}_y, r\mathbf{I}_T)$. Moreover, since $y \in X$, we have $\mathbf{v}_X \in \mathsf{span}(\mathbf{M}_y)$. Hence, $(r,0,\ldots,0)$ is also not in the row span of

$$\begin{pmatrix} r\mathbf{M}_y^\top \\ s\mathbf{v}_X^\top \end{pmatrix} = \begin{pmatrix} r\mathbf{m}_y & r\mathbf{I}_T \\ s\mathbf{v}_{X,0}^\top & s\mathbf{v}_{X,1\to T}^\top \end{pmatrix}$$

Hence $(1,0,\ldots,0)$ is not in the row span of the matrix on the left-hand side. Therefore, α is completely hidden.

This concludes the proof. ☐

6.8 CP-ABE with Constant-Size Ciphertexts

CP-ABE with Constant-Size Ciphertexts. We next construct a PES for the predicate $\mathsf{P}^{\mathsf{CP\text{-}MBF\text{-}31}}$ (bounded formula sizes and attribute sets, but large-universe) with constant-size ciphertext encodings.[5] We achieve this by the following two lemma.

Lemma 18. $\mathsf{P}^{\mathsf{CP\text{-}MBF\text{-}63}}$ *can be embedded into* $\mathsf{P}^{\mathsf{KP\text{-}MBF\text{-}63}}$.

We use the *depth-universal circuit* of Cook and Hoover [17], where we recapitulate in the following proposition. It implies a universal circuit for NC1 (log-depth circuits), or equivalently Boolean formulae.

Proposition 1 ([17]). *For any k, M, D there is a universal circuit $\mathsf{U}_{k,M,D}$ that can simulate any circuit C having k inputs, of size at most M and depth at most D, and $\mathsf{U}_{k,M,D}$ has depth $\mathsf{D}_D = O(D)$ and size $\mathsf{S}_{M,D} = O(M^3 D/\log M)$. The input to the circuit $\mathsf{U}_{k,M,D}$ consists of the regular input k bits and $\mathsf{C}_M = O(M^2 \log M)$ bits representing the description of the simulated circuit C.*

[5] Exactly the same bound requirements are analogously assumed for the previous CP-ABE schemes (for monotone span programs) with constant-size ciphertexts [1,9].

Proof (of Lemma 18). Consider $\mathsf{P}^{\text{CP-MBF-63}}_{\kappa} : \mathcal{Y}_{\kappa} \times \mathcal{X}_{\kappa} \to \{0,1\}$ and $\mathsf{P}^{\text{KP-MBF-63}}_{\kappa'} : \mathcal{X}_{\kappa'} \times \mathcal{Y}_{\kappa'} \to \{0,1\}$ (see Definition 17). We assume w.l.o.g. that $\mathcal{U} = [u]$. We map $\kappa = (U, T, N, M, D, \varphi) \mapsto \kappa' = (U', T', N', M', D', \varphi')$ by setting $U' = T' = N' = \mathsf{C}_{\tilde{M}}$, $M' = \mathsf{S}_{\tilde{M},\tilde{D}}$, $D' = \mathsf{D}_{\tilde{D}}$, $\varphi' = 1$, where $\tilde{M} = O(M + U), \tilde{D} = O(D + \log U + \log \varphi)$ are set as below.

Towards using the universal circuit, we first map the ciphertext attribute $Y = (\mathbf{y}, f) \in \mathcal{Y}_{\kappa}$ of CP-ABE (which consists of the input label $\mathbf{y} \in \mathcal{U}^n$ and a boolean formula $f : \{0,1\}^n \to \{0,1\}$) to its corresponding boolean formula $g_Y : \{0,1\}^{\varphi U+1} \to \{0,1\}$ (now with *globally-fixed* input labels), and map the key attribute $X \in \mathcal{X}_{\kappa}$ of CP-ABE to its corresponding bit string $\mathbf{b}_X \in \{0,1\}^{\varphi U+1}$, so that we will have

$$g_Y(\mathbf{b}_X) = \mathsf{P}^{\text{CP-ABE-63}}_{\kappa}(Y, X). \tag{6}$$

This can be done as follows. Consider a new universe $\mathcal{U}' = [U] \times [\varphi] \cup \{\mathsf{dummy}\}$ and assume some lexicographical order in $\mathcal{U}'$.

- From $\mathbf{y} \in \mathcal{U}^n$, we define $\mathbf{y}' \in (\mathcal{U}')^n$ by setting $y'_j = (y_j, q)$ if y_j is the q-th appearance of the same attribute in $\mathbf{y}$, *i.e.*, $q = |\{\iota \in [j] | y_\iota = y_j\}|$. We construct g_Y to be the same as f except with the following modifications. First the input labels are modified from $\mathbf{y}$ to $\mathbf{y}'$. Then we re-order the input wires so that they are in a lexicographical order in $\mathcal{U}'$. We next add an input wire labelled dummy. Then for all $b \in \mathcal{U}'$ where x does not appear in $\mathbf{y}'$, we add an input wire labelled and take an AND over them and the dummy input wire then take an OR over the output of this AND gate and the output wire, and output it as a new output wire. We expand the AND gate into many AND gates with fan-in 2 in depth $O(\log \varphi U)$. This completes the specification of g_Y, which always have all input wires labelled fully by $\mathcal{U}'$ (in the lexicographical order). By inspection, g_Y has depth $\tilde{D} = O(D + \log U + \log \varphi)$ and size $\tilde{M} = O(M + U)$.
- For $X \in \mathcal{X}_{\kappa}$, we set $\mathbf{b}_X = (b_j)_{j \in \mathcal{U}'} \in \{0,1\}^{\varphi U+1}$ (in the lexicographical order) as follows. Let $b_{\mathsf{dummy}} = 0$. For each $j \in \mathcal{U}' \setminus \{\mathsf{dummy}\}$ we parse $j = (\iota, q)$ and for all $q \in [\varphi]$ we set $b_j = 1$ iff $\iota \in X$.

It is straightforward to see that Eq. (6) holds.

Now that we have the set of globally fixed input labels, we can use a depth-universal circuit (Proposition 1) $\mathsf{U} := \mathsf{U}_{\varphi U+1, \tilde{M}, \tilde{D}}$. For a boolean formula $g_Y : \{0,1\}^{\varphi U+1} \to \{0,1\}$, we write its description as a bit string $\mathsf{desc}(g_Y) \in \{0,1\}^{\mathsf{C}_{\tilde{M}}}$ where $\mathsf{C}_{\tilde{M}} = O(\tilde{M}^2 \log \tilde{M})$ (by using the extended encoding in [17]). The universal circuit $\mathsf{U} : \{0,1\}^{\mathsf{C}_{\tilde{M}}} \times \{0,1\}^{\varphi U+1} \to \{0,1\}$ has the following property:

$$\mathsf{U}(\mathsf{desc}(g_Y), \mathbf{b}_X) = g_Y(\mathbf{b}_X). \tag{7}$$

We then view $\mathsf{U}(\cdot, \mathbf{b}_X)$ as a boolean formula $h_X : \{0,1\}^{\mathsf{C}_{\tilde{M}}} \to \{0,1\}$ with the input labels being $1, \ldots, \mathsf{C}_{\tilde{M}}$. This yields

$$h_X(\mathsf{desc}(g_Y)) = \mathsf{U}(\mathsf{desc}(g_Y), \mathbf{b}_X) \tag{8}$$

We can finally map

$$\begin{aligned} Y &\mapsto A_Y = \{\iota \in [\mathsf{C}_{\tilde{M}}] \,|\, \text{The } \iota\text{-th element in } \mathsf{desc}(g_Y) \text{ is } 1 \ \} \in \mathcal{X}'_{\kappa}, \\ X &\mapsto B_X = ([\mathsf{C}_{\tilde{M}}], h_X) \in \mathcal{Y}'_{\kappa}, \end{aligned}$$

where we consider the attribute universe $[\mathsf{C}_{\tilde{M}}]$ in KP-ABE. From the definition of $\mathsf{P}^{\text{KP-MBF-63}}_{\kappa'}$ we have

$$\mathsf{P}^{\text{KP-MBF-63}}_{\kappa'}(A_Y, B_X) = h_X(\mathsf{desc}(g_Y)) \tag{9}$$

From Eq. (6),Eq. (7),Eq. (8),Eq. (9), we thus have

$$\mathsf{P}^{\text{CP-MBF-63}}_{\kappa}(Y, X) = 1 \Leftrightarrow \mathsf{P}^{\text{KP-MBF-63}}_{\kappa'}(A_Y, B_X) = 1.$$

This concludes the lemma. $\square$

Lemma 19. $\mathsf{P}^{\text{CP-MBF-31}}$ *can be embedded into* $\mathsf{P}^{\text{CP-MBF-63}}$.

Proof. Consider $\mathsf{P}^{\text{CP-MBF-31}}_{\kappa} : \mathcal{Y}_{\kappa} \times \mathcal{X}_{\kappa} \to \{0,1\}$ and $\mathsf{P}^{\text{CP-MBF-63}}_{\kappa'} : \mathcal{Y}'_{\kappa'} \times \mathcal{X}'_{\kappa'} \to \{0,1\}$ and (*cf.* the key-policy version of definition in Definition 17.). We map $\kappa = (\text{-}, T, N, M, D, \varphi) \mapsto \kappa' = (U', T', N', M', D', \varphi')$ as follows. In the large universe CP-ABE, the universe is $\mathbb{Z}_p$ where $\lambda = \lceil \log p \rceil$ is the security parameter. We set $U' = 2T\lambda$, $T' = T\lambda$, $N' = NT\lambda$, $M' = M + 2T\lambda - 1$, $D' = D + \lceil \log T \rceil + \lceil \log \lambda \rceil$, and $\varphi' = \varphi$. We set a new universe as $\mathcal{U}' = [T] \times [\lambda] \times \{0,1\}$. We map as follows.

- From $Y = (\mathbf{y}, f) \in \mathcal{Y}_{\kappa}$, we construct $Y' = (\mathbf{y}', f') \in \mathcal{Y}'_{\kappa'}$ as follows. Parse $\mathbf{y} = (y_1, \ldots, y_n)$. For each y_j in $\mathbf{y}$, denote the bit decomposition of y_j as $(y_{j,1}, \ldots, y_{j,\lambda}) \in \{0,1\}^{\lambda}$. For $k \in [T], j \in [n]$, define

$$\mathbf{y}_{k,j} := ((k, 1, y_{j,1}), \ldots, (k, \lambda, y_{j,\lambda})) \in (\mathcal{U}')^{\lambda}.$$

  We set the new input wires $\mathbf{y}'$ as

$$\mathbf{y}' = \mathbf{y}_{1,1} || \cdots || \mathbf{y}_{1,i} || \mathbf{y}_{2,1} || \cdots || \mathbf{y}_{2,i} || \cdots || \mathbf{y}_{T,1} || \cdots || \mathbf{y}_{T,i},$$

  where $||$ is the concatenation. Note that if the maximum repetition in $\mathbf{y}$ is φ, then the maximum repetition in $\mathbf{y}'$ is also φ (hence we set $\varphi' = \varphi$). We construct f' to be exactly the same as f except that for each input wire y_j of f, we add the following (depth-2, large fan-in) sub-circuit:

$$\bigvee_{k=1}^{T} \bigwedge_{\iota=1}^{\lambda} (k, \iota, y_{j,\iota})$$

  in such a way that its output wire connects to the wire y_j. Note that now we have that the input labels of f' are exactly $\mathbf{y}'$. This sub-circuit can be then straightforwardly converted to an equivalent fan-in-2 circuit with depth $\lceil \log T \rceil + \lceil \log \lambda \rceil$ (by expanding the OR and the AND). As a result, f' is a Boolean formulae with depth $D' = D + \lceil \log T \rceil + \lceil \log \lambda \rceil$, size $M' = M + 2T\lambda - 1$, and input length $nT\lambda$ (recall that D, M is the depth and the size of f, respectively).
- From $X \in \mathcal{X}_{\kappa}$, we construct $X' \in \mathcal{X}'_{\kappa'}$ as follows. Parse $X = \{x_1, \ldots, x_t\}$. For each $x_j \in X$, denote the bit decomposition of x as $(x_{j,1}, \ldots, x_{j,\lambda})$. We set

$$X' = \bigcup_{j=1}^{t} \{(j, 1, x_{j,1}), \ldots, (j, \lambda, x_{j,\lambda})\}.$$

  Note that we have $X' \in \binom{\mathcal{U}'}{\le T'}$, where we let $T' = T\lambda$. This is since $t \le T$.

We claim that

$$\mathsf{P}^{\text{CP-MBF-31}}_{\kappa}(Y, X) = \mathsf{P}^{\text{CP-MBF-63}}_{\kappa'}(Y', X').$$

This holds since the functionality of the above sub-circuit is to compute the satisfiability for $y_j \in X$. $\square$

Combining both lemmata, we obtain the following corollary.

Corollary 3. $\mathsf{P}^{\text{CP-MBF-31}}$ *can be embedded into* $\mathsf{P}^{\text{KP-MBF-63}}$.

We can inspect the efficiency of the resulting large-universe CP-ABE from the small-universe KP-ABE by combining the parameter mappings in the proofs of both lemmata. That is, if we let κ and κ'' be the indexes of the large-universe CP-ABE and the small-universe KP-ABE, respectively, then via the two lemmata we have the combined map that takes $\kappa = (\text{-}, T, N, M, D, \varphi) \mapsto \kappa'' = (U'', T'', N'', M'', D'', \varphi'')$ where $\varphi'' = 1$ and

$$\begin{aligned} U'' = T'' = N'' &= O((M + T\lambda)^2 \log(M + T\lambda)) = \tilde{O}((M + T\lambda)^2), \\ M'' &= O((M + T\lambda)^3 (D + \log T + \log \lambda + \log \varphi)/\log(M + T\lambda)), \\ D'' &= O(D + \log T + \log \lambda + \log \varphi). \end{aligned}$$

Applying the combined conversion in this sub section to the KP-ABE with constant-size ciphertexts of the previous subsection (§6.7), we obtain the large-universe CP-ABE with constant-size ciphertexts. The public key size is $O(T'') = \tilde{O}((M + T\lambda)^2)$, while the secret key size is $O(N''T'') = \tilde{O}((M + T\lambda)^4)$.

6.9 KP-ABE, CP-ABE with Constant-Size Keys

We apply the dual conversion to ABE with constant-size ciphertexts to obtain the following.

(Non-monotone) CP-ABE with Constant-Size Keys. This can be obtained by applying the dual conversion to the (non-monotone) KP-ABE with constant-size ciphertexts of §6.7. The property of constant-size ciphertexts in KP-ABE becomes the property of constant-size keys in CP-ABE since the size of ciphertexts preserve to the size of keys (and vice versa) via the dual conversion.

KP-ABE with Constant-Size Keys. This can be obtained by applying the dual conversion to the CP-ABE with constant-size ciphertexts of §6.8.

Acknowledgement. Nuttapong Attrapadung was partly supported by JST CREST Grant Number JPMJCR19F6, and by JSPS KAKENHI Kiban-A Grant Number 19H01109.

References

1. S. Agrawal and M. Chase. A study of pair encodings: Predicate encryption in prime order groups. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 259–288. Springer, Heidelberg, Jan. 2016.
2. S. Agrawal and M. Chase. FAME: Fast attribute-based message encryption. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 17*, pages 665–682. ACM Press, Oct. / Nov. 2017.
3. S. Agrawal and M. Chase. Simplifying design and analysis of complex predicate encryption schemes. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 627–656. Springer, Heidelberg, Apr. / May 2017.
4. S. Agrawal, M. Maitra, and S. Yamada. Attribute based encryption (and more) for nondeterministic finite automata from LWE. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2019, Part II*, LNCS, pages 765–797. Springer, Heidelberg, Aug. 2019.
5. M. Ambrona, G. Barthe, and B. Schmidt. Generic transformations of predicate encodings: Constructions and applications. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 36–66. Springer, Heidelberg, Aug. 2017.
6. N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.
7. N. Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, Dec. 2016.
8. N. Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In V. Rijmen and Y. Ishai, editors, *EUROCRYPT 2019, Part I*, LNCS, pages 34–67. Springer, Heidelberg, May 2019.
9. N. Attrapadung, G. Hanaoka, and S. Yamada. Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 575–601. Springer, Heidelberg, Nov. / Dec. 2015.
10. N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 90–108. Springer, Heidelberg, Mar. 2011.
11. N. Attrapadung and S. Yamada. Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings. In K. Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 87–105. Springer, Heidelberg, Apr. 2015.
12. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, Aug. 2001.
13. D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 455–470. Springer, Heidelberg, Dec. 2008.
14. J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, Apr. 2015.

15. J. Chen, J. Gong, L. Kowalczyk, and H. Wee. Unbounded ABE via bilinear entropy expansion, revisited. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 503–534. Springer, Heidelberg, Apr. / May 2018.
16. J. Chen and H. Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In M. Abdalla and R. D. Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, Sept. 2014.
17. S. Cook and H. Hoover. A depth-universal circuit. *SIAM Journal on Computing*, 14(4):833–839, 1985.
18. C. Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In K. Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 200–215. Springer, Heidelberg, Dec. 2007.
19. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, Jan. 2017.
20. J. Gong, X. Dong, J. Chen, and Z. Cao. Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 624–654. Springer, Heidelberg, Dec. 2016.
21. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.
22. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06*, pages 89–98. ACM Press, Oct. / Nov. 2006. Available as Cryptology ePrint Archive Report 2006/309.
23. Z. Jafargholi, C. Kamath, K. Klein, I. Komargodski, K. Pietrzak, and D. Wichs. Be adaptive, avoid overcommitting. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 133–163. Springer, Heidelberg, Aug. 2017.
24. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, Apr. 2008.
25. L. Kowalczyk and H. Wee. Compact adaptively secure ABE for $\mathsf{NC}s^1$ from k-lin. In V. Rijmen and Y. Ishai, editors, *EUROCRYPT 2019, Part I*, LNCS, pages 3–33. Springer, Heidelberg, May 2019.
26. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, Heidelberg, May / June 2010.
27. A. B. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011.
28. T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Heidelberg, Aug. 2010.
29. T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, Dec. 2012.
30. R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 195–203. ACM Press, Oct. 2007.
31. Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13*, pages 463–474. ACM Press, Nov. 2013.
32. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.
33. K. Takashima. Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption. In M. Abdalla and R. D. Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 298–317. Springer, Heidelberg, Sept. 2014.
34. J. Tomida, Y. Kawahara, and R. Nishimaki. Fast, compact, and expressive attribute-based encryption. Cryptology ePrint Archive, Report 2019/966, 2019. https://eprint.iacr.org/2019/966.
35. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, Aug. 2009.
36. B. Waters. Functional encryption for regular languages. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Heidelberg, Aug. 2012.

37. H. Wee. Dual system encryption via predicate encodings. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, Feb. 2014.
38. S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro. A framework and compact constructions for non-monotonic attribute-based encryption. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 275–292. Springer, Heidelberg, Mar. 2014.

A Concrete Descriptions of Our Instantiations

For self-containment, as examples, we provide the concrete descriptions of three schemes that are newly obtained by our framework, namely, completely unbounded KP/CP-ABE for monotone Boolean formulae and KP-ABE with constant-size ciphertexts for monotone Boolean formulae.

A.1 Completely Unbounded KP-ABE for Monotone Formulae

A predicate for completely unbounded KP-ABE for monotone Boolean formulae, $\mathsf{P}^{\text{KP-MBF}} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, is defined as follows:

- $\mathcal{X} = 2^{\mathbb{Z}_p}$.
- $\mathcal{Y} = \bigcup_{i \in \mathbb{N}} (\mathbb{Z}_p^i \times \mathcal{F}_i)$, where $\mathcal{F}_i$ consists of all monotone Boolean formulae with input length i.
- For $X = \{x_1, \ldots, x_t\} \in \mathcal{X}$ and $Y = ((y_1, \ldots, y_n), f) \in \mathcal{Y}$ where $f : \{0,1\}^n \to \{0,1\}$, we set $b_i = 1$ iff $y_i \in X$. Then, we define $\mathsf{P}^{\text{KP-MBF}}(X, Y) = 1 \Leftrightarrow f(b_1, \ldots, b_n) = 1$.

In our framework, the predicate $\mathsf{P}^{\text{KP-MBF}}$ can be embedded into

$$\mathsf{KBF1}[\mathsf{P}^{\mathsf{IBBE}}] = \mathsf{KBF1}[\mathsf{Dual}[\mathsf{KBF1}_{\mathsf{OR}}[\mathsf{P}^{\mathsf{IBE}}]]].$$

Thus, the corresponding PES $\Gamma^{\text{KP-MBF}}$ is described as follows:

- $\mathsf{Param}() = 3$
- $\mathsf{EncCt}(\{x_1, \ldots, x_t\}) = (t, 0, \mathbf{c})$. Polynomials $\mathbf{c} = (c_1, \ldots, c_t)$ are defined as follows:

$$c_i = s_0 w_1 - s_i(x_i w_2 + w_3).$$

- $\mathsf{EncKey}((y_1, \ldots, y_n), f) = (n, \tau, \mathbf{k})$, where τ is the number of AND gates in f. Let $\mathsf{Share_p}$ be the algorithm defined in Fig 4. Polynomials $\mathbf{k} = (k_{1,1}, \ldots, k_{1,n}, k_{2,1}, \ldots, k_{2,n})$ are defined as follows:

$$\sigma_1, \ldots, \sigma_n \leftarrow \mathsf{Share_p}(f, \alpha, \hat{\mathbf{r}}_{-\alpha} = (\hat{r}_1, \ldots, \hat{r}_\tau)),$$
$$k_{1,i} = \sigma_i - r_i w_1, \quad k_{2,i} = r_i(y_i w_2 + w_3).$$

- $\mathsf{Pair}(\{x_1, \ldots, x_t\}, (y_1, \ldots, y_n), f) = (\mathbf{E}, \overline{\mathbf{E}})$. Let $S \subseteq \{i \mid y_i \in \{x_1, \ldots, x_t\}\}$ be the set such that $\sum_{i \in S} \sigma_i = \alpha$. $\mathbf{E} = (e_{i,j})_{i \in [t+1], j \in [2n]}$ and $\overline{\mathbf{E}} = (\bar{e}_{i,j})_{i \in [t], j \in [n]}$ are defined as the following equation holds:

$$\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \sum_{i \in S} (s_0 k_{1,i} + c_{\phi(i)} r_i + s_{\phi(i)} k_{2,i}),$$

where $\phi : S \to [t]$ is a function such that $y_i = x_{\phi(i)}$.

Because $\Gamma^{\text{KP-MBF}}$ is obtained by applying the dual conversion once to a PES with single-variable PMH, namely, Γ^{IBE}, $\Gamma^{\text{KP-MBF}}$ satisfies $(2,2)$-$\mathsf{KE\text{-}ind}$. Thus, the concrete scheme is described as follows.

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

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \;\; \mathbf{A}, \mathbf{B} \leftarrow \mathbb{Z}_p^{(k+2) \times k}, \;\; \mathbf{h} \leftarrow \mathbb{Z}_p^{k+2}, \;\; \mathbf{W}_1, \ldots, \mathbf{W}_3 \leftarrow \mathbb{Z}_p^{(k+2) \times (k+2)}$$
$$\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_3^\top \mathbf{A}]_1, [\mathbf{A}^\top \mathbf{h}]_\mathsf{T})$$
$$\mathsf{msk} = (\mathbf{B}, \mathbf{h}, \mathbf{W}_1, \ldots, \mathbf{W}_3).$$

$\mathsf{Enc}(\mathsf{pk}, X, M)$**:** It takes pk, an attribute $X = \{x_1, \ldots, x_t\} \in \mathcal{X}$, and a message $M \in G_\mathsf{T}$ and outputs ct_X as follows. Let Share be an algorithm defined in Fig 5.

$$\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_t \leftarrow \mathbb{Z}_p^k$$
$$\mathsf{ct}_{1,i} = [\mathbf{A}\mathbf{s}_i]_1,\ \mathsf{ct}_{2,i} = [\mathbf{W}_1^\top \mathbf{A}\mathbf{s}_0 - (x_i \mathbf{W}_2^\top + \mathbf{W}_3^\top)\mathbf{A}\mathbf{s}_i]_1,\ \mathsf{ct}_3 = [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{h}]_\mathsf{T} M$$
$$\mathsf{ct}_X = (\{\mathsf{ct}_{1,i}\}_{i\in[t]^+}, \{\mathsf{ct}_{2,i}\}_{i\in[t]}, \mathsf{ct}_3).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, Y)$**:** It takes pk, msk, and a set $Y = ((y_1, \ldots, y_n), f) \in \mathcal{Y}$ and outputs sk_Y as follows.

$$\mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^k,\ \ \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n \leftarrow \mathsf{Share}(f, \mathbf{h})$$
$$\mathsf{sk}_{1,i} = [\mathbf{B}\mathbf{r}_i]_2,\ \ \mathsf{sk}_{2,i} = [\boldsymbol{\sigma}_i - \mathbf{W}_1\mathbf{B}\mathbf{r}_i]_2,\ \ \mathsf{sk}_{3,i} = [(y_i\mathbf{W}_2 + \mathbf{W}_3)\mathbf{B}\mathbf{r}_i]_2$$
$$\mathsf{sk}_Y = \{\mathsf{sk}_{1,i}, \mathsf{sk}_{2,i}, \mathsf{sk}_{3,i}\}_{i\in[n]}.$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_X, \mathsf{sk}_Y)$**:** It takes pk, $\mathsf{ct}_X = (\{\mathsf{ct}_{1,i}\}_{i\in[t]^+}, \{\mathsf{ct}_{2,i}\}_{i\in[t]}, \mathsf{ct}_3)$, and $\mathsf{sk}_Y = \{\mathsf{sk}_{1,i}, \mathsf{sk}_{2,i}, \mathsf{sk}_{3,i}\}_{i\in[n]}$ such that $\mathsf{P}^{\mathsf{KP\text{-}MBF}}(X, Y) = 1$. Let S and ϕ be the same as those defined in Pair of $\Gamma^{\mathsf{KP\text{-}MBF}}$. It outputs M' as follows.

$$M' = \mathsf{ct}_3 \Big/ \prod_{i\in S} e(\mathsf{ct}_{1,0}, \mathsf{sk}_{2,i}) e(\mathsf{ct}_{2,\phi(i)}, \mathsf{sk}_{1,i}) e(\mathsf{ct}_{1,\phi(i)}, \mathsf{sk}_{3,i}).$$

A.2 Completely Unbounded CP-ABE for Monotone Formulae

A predicate for completely unbounded CP-ABE for monotone Boolean formulae, $\mathsf{P}^{\mathsf{CP\text{-}MBF}} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, is defined as follows:

- $\mathcal{X} = \bigcup_{i\in\mathbb{N}} (\mathbb{Z}_p^i \times \mathcal{F}_i)$, where $\mathcal{F}_i$ consists of all monotone Boolean formulae with input length i.
- $\mathcal{Y} = 2^{\mathbb{Z}_p}$.
- For $X = ((x_1, \ldots, x_n), f) \in \mathcal{X}$ and $Y = \{y_1, \ldots, y_t\} \in \mathcal{Y}$ where $f : \{0,1\}^n \to \{0, 1\}$, we set $b_i = 1$ iff $x_i \in Y$. Then, we define $\mathsf{P}^{\mathsf{CP\text{-}MBF}}(X, Y) = 1 \Leftrightarrow f(b_1, \ldots, b_n) = 1$.

In our framework, the predicate $\mathsf{P}^{\mathsf{CP\text{-}MBF}}$ can be embedded into

$$\mathsf{Dual}[\mathsf{KBF1}[\mathsf{P}^{\mathsf{IBBE}}]] = \mathsf{Dual}[\mathsf{KBF1}[\mathsf{Dual}[\mathsf{KBF1}_{\mathsf{OR}}[\mathsf{P}^{\mathsf{IBE}}]]]].$$

Thus, the corresponding PES $\Gamma^{\mathsf{CP\text{-}MBF}}$ is described as follows:

- $\mathsf{Param}() = 4$
- $\mathsf{EncCt}((x_1, \ldots, x_n), f) = (n, \tau, \mathbf{c})$, where τ is the number of AND gates in f. Let $\mathsf{Share}_\mathsf{p}$ be the algorithm defined in Fig 4. Polynomials $\mathbf{c} = (c_{1,1}, \ldots, c_{1,n}, c_{2,1}, \ldots, c_{2,n})$ are defined as follows:

$$\sigma_1, \ldots, \sigma_n \leftarrow \mathsf{Share}_\mathsf{p}(f, s_0 w_1, \hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_\tau)),$$
$$c_{1,i} = \sigma_i - s_i w_2,\ \ c_{2,i} = s_i(x_i w_3 + w_4).$$

- $\mathsf{EncKey}(\{y_1, \ldots, y_t\}) = (t + 1, 0, \mathbf{k})$. Polynomials $\mathbf{k} = (k_0, k_1, \ldots, k_t)$ are defined as follows:

$$k_0 = \alpha - r_0 w_1,\ \ k_i = r_0 w_2 - r_i(y_i w_3 + w_4).$$

- $\mathsf{Pair}((x_1, \ldots, x_n), f, \{y_1, \ldots, y_t\}) = (\mathbf{E}, \overline{\mathbf{E}})$. Let $S \subseteq \{i \mid x_i \in \{y_1, \ldots, y_t\}\}$ be the set such that $\sum_{i\in S} \sigma_i = s_0 w_1$. $\mathbf{E} = (e_{i,j})_{i\in[n+1], j\in[t+1]}$ and $\overline{\mathbf{E}} = (\bar{e}_{i,j})_{i\in[2n], j\in[t+1]}$ are defined as the following equation holds:

$$\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = s_0 k_0 + \sum_{i\in S} (s_i k_{\phi(i)} + c_{1,i} r_0 + c_{2,i} r_{\phi(i)}),$$

where $\phi : S \to [t]$ is a function such that $x_i = y_{\phi(i)}$.

Because $\Gamma^{\mathsf{CP\text{-}MBF}}$ is obtained by applying the dual conversion twice to a PES with single-variable PMH, namely, Γ^{IBE}, $\Gamma^{\mathsf{CP\text{-}MBF}}$ satisfies $(3,3)$-KE-ind. Thus, the concrete scheme is described as follows.

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

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \ \ \mathbf{A}, \mathbf{B} \leftarrow \mathbb{Z}_p^{(k+3)\times k}, \ \ \mathbf{h} \leftarrow \mathbb{Z}_p^{k+3}, \ \ \mathbf{W}_1, \dots, \mathbf{W}_4 \leftarrow \mathbb{Z}_p^{(k+3)\times(k+3)} \\
&\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \dots, [\mathbf{W}_4^\top \mathbf{A}]_1, [\mathbf{A}^\top \mathbf{h}]_\mathsf{T}) \\
&\mathsf{msk} = (\mathbf{B}, \mathbf{h}, \mathbf{W}_1, \dots, \mathbf{W}_4).
\end{aligned}$$

$\mathsf{Enc}(\mathsf{pk}, X, M)$**:** It takes pk, an attribute $X = ((x_1, \dots, x_n), f) \in \mathcal{X}$, and a message $M \in G_\mathsf{T}$ and outputs ct_X as follows. Let Share be an algorithm defined in Fig 5.

$$\begin{aligned}
&\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_n \leftarrow \mathbb{Z}_p^k, \ \ \boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n \leftarrow \mathsf{Share}(f, \mathbf{W}_1^\top \mathbf{A}\mathbf{s}_0) \\
&\mathsf{ct}_{1,i} = [\mathbf{A}\mathbf{s}_i]_1, \ \ \mathsf{ct}_{2,i} = [\boldsymbol{\sigma}_i - \mathbf{W}_2^\top \mathbf{A}\mathbf{s}_i]_1, \ \ \mathsf{ct}_{3,i} = [(x_i \mathbf{W}_3^\top + \mathbf{W}_4^\top)\mathbf{A}\mathbf{s}_i]_1 \\
&\mathsf{ct}_4 = [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{h}]_\mathsf{T} M \\
&\mathsf{ct}_X = (\{\mathsf{ct}_{1,i}\}_{i\in[n]^+}, \{\mathsf{ct}_{2,i}, \mathsf{ct}_{3,i}\}_{i\in[n]}, \mathsf{ct}_4).
\end{aligned}$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, Y)$**:** It takes pk, msk, and a set $Y = \{y_1, \dots, y_t\} \in \mathcal{Y}$ and outputs sk_Y as follows.

$$\begin{aligned}
&\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_t \leftarrow \mathbb{Z}_p^k \\
&\mathsf{sk}_{1,i} = [\mathbf{B}\mathbf{r}_i]_2, \ \ \mathsf{sk}_2 = [\mathbf{h} - \mathbf{W}_1\mathbf{B}\mathbf{r}_0]_2, \ \ \mathsf{sk}_{3,i} = [\mathbf{W}_2\mathbf{B}\mathbf{r}_0 - (y_i\mathbf{W}_3 + \mathbf{W}_4)\mathbf{B}\mathbf{r}_i]_2 \\
&\mathsf{sk}_Y = (\{\mathsf{sk}_{1,i}\}_{i\in[t]^+}, \mathsf{sk}_2, \{\mathsf{sk}_{3,i}\}_{i\in[t]}).
\end{aligned}$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_X, \mathsf{sk}_Y)$**:** It takes pk, $\mathsf{ct}_X = (\{\mathsf{ct}_{1,i}\}_{i\in[n]^+}, \{\mathsf{ct}_{2,i}, \mathsf{ct}_{3,i}\}_{i\in[n]}, \mathsf{ct}_4)$, and $\mathsf{sk}_Y = (\{\mathsf{sk}_{1,i}\}_{i\in[t]^+}, \mathsf{sk}_2, \{\mathsf{sk}_{3,i}\}_{i\in[t]})$ such that $\mathsf{P}^{\mathsf{CP\text{-}MBF}}(X, Y) = 1$. Let S and ϕ be the same as those defined in Pair of $\Gamma^{\mathsf{CP\text{-}MBF}}$. It outputs M' as follows.

$$M' = \mathsf{ct}_4 \Big/ e(\mathsf{ct}_{1,0}, \mathsf{sk}_2) \prod_{i\in S} e(\mathsf{ct}_{1,i}, \mathsf{sk}_{3,\phi(i)}) e(\mathsf{ct}_{2,i}, \mathsf{sk}_{1,0}) e(\mathsf{ct}_{3,i}, \mathsf{sk}_{1,\phi(i)}).$$

A.3 KP-ABE with Constant-Size Ciphertexts for Monotone Formulae

Following §6.7, we consider ABE for $\mathsf{P}^{\mathsf{KP\text{-}MBF\text{-}16}} = \mathsf{KBF1}[\mathsf{P}^{\mathsf{IBBE}'}]$. The corresponding PES $\Gamma^{\mathsf{KP\text{-}MBF\text{-}16}}$ is described as follows:

- $\mathsf{Param}(T) = T + 2$
- $\mathsf{EncCt}(X) = (0, 0, c)$ where $c = s\mathbf{w}^\top \mathbf{v}'_X$, where the non-lone variable is s.
- $\mathsf{EncKey}((y_1, \dots, y_n), f) = (n, \tau, \mathbf{k})$, where τ is the number of AND gates in f. Let $\mathsf{Share}_\mathsf{p}$ be the algorithm defined in Fig 4. Polynomials $\mathbf{k} = (\mathbf{k}^{(1)}, \dots, \mathbf{k}^{(n)})$ are defined as follows:

$$\begin{aligned}
&\sigma_1, \dots, \sigma_n \leftarrow \mathsf{Share}_\mathsf{p}(f, \alpha, \hat{\mathbf{r}} = (\hat{r}_1, \dots, \hat{r}_\tau)), \\
&\mathbf{k}^{(i)} = (\sigma_i, \mathbf{0}) + r_i \mathbf{w}^\top \mathbf{M}'_{y_i}.
\end{aligned}$$

- $\mathsf{Pair}(X, (y_1, \dots, y_n), f) = (\mathbf{e}, \bar{\mathbf{e}})$. Let $S \subseteq \{i \mid y_i \in X\}$ be the set such that $\sum_{i\in S} \sigma_i = \alpha$. Then, $\mathbf{e} = (e_1, \dots, e_{n(T+1)})$ and $\bar{\mathbf{e}} = (\bar{e}_1, \dots, \bar{e}_n)$ are defined as the following equation holds:

$$s\mathbf{e}\mathbf{k}^\top + c\bar{\mathbf{e}}\mathbf{r}^\top = \sum_{i\in S} (s\mathbf{k}^{(i)}\mathbf{v}''_X - r_i c),$$

  where $\mathbf{v}''_X = (1, a_1, \dots, a_T)^\top$.

Note that a_i, $\mathbf{v}'_X$, $\mathbf{v}''_X$ and $\mathbf{M}'_y$ is defined the same as those in §6.7, and we have $\mathbf{M}'_y\mathbf{v}''_X = \mathbf{v}'_X$ iff $y \in X$.

Because $\Gamma^{\mathsf{KP\text{-}MBF\text{-}16}}$ is obtained without the dual conversion, $\Gamma^{\mathsf{KP\text{-}MBF\text{-}16}}$ satisfies $(1,1)$-$\mathsf{KE\text{-}ind}$. Thus, the concrete scheme is described as follows.

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

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda),\ \ \mathbf{A}, \mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k},\ \ \mathbf{h} \leftarrow \mathbb{Z}_p^{k+1},\ \ \mathbf{W}_1, \ldots, \mathbf{W}_{T+2} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}\\
&\mathsf{pk} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{W}_1^\top\mathbf{A}]_1, \ldots, [\mathbf{W}_{T+2}^\top\mathbf{A}]_1, [\mathbf{A}^\top\mathbf{h}]_\mathsf{T})\\
&\mathsf{msk} = (\mathbf{B}, \mathbf{h}, \mathbf{W}_1, \ldots, \mathbf{W}_{T+2}).
\end{aligned}$$

$\mathsf{Enc}(\mathsf{pk}, X, M)$**:** It takes pk, an attribute $X \in \mathcal{X}_\kappa$, and a message $M \in G_\mathsf{T}$ and outputs ct_X as follows.

$$\begin{aligned}
&\mathbf{s} \leftarrow \mathbb{Z}_p^k,\ \ \mathsf{ct}_1 = [\mathbf{A}\mathbf{s}]_1,\ \ \mathsf{ct}_2 = \left[\sum_{\ell\in[T+2]} v'_{X,\ell}\mathbf{W}_\ell\mathbf{A}\mathbf{s}\right]_1,\ \ \mathsf{ct}_3 = [\mathbf{s}^\top\mathbf{A}^\top\mathbf{h}]_\mathsf{T}M\\
&\mathsf{ct}_X = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3),
\end{aligned}$$

where $v'_{X,\ell}$ denotes the ℓ-th element of $\mathbf{v}'_X$.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, Y)$**:** It takes pk, msk, and a predicate $Y = ((y_1, \ldots, y_n), f) \in \mathcal{Y}_\kappa$ and outputs sk_Y as follows. Let Share be an algorithm defined in Fig 5.

$$\begin{aligned}
&\mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^k,\ \ \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n \leftarrow \mathsf{Share}(f, \mathbf{h})\\
&\mathsf{sk}_{1,i} = [\mathbf{B}\mathbf{r}_i]_2,\ \ \mathsf{sk}_{2,i,j} = \begin{cases} [\boldsymbol{\sigma}_i + \sum_{\ell\in[T+2]} m'_{y_i,\ell,j}\mathbf{W}_\ell\mathbf{B}\mathbf{r}_i]_2 & j = 1\\ [\sum_{\ell\in[T+2]} m'_{y_i,\ell,j}\mathbf{W}_\ell\mathbf{B}\mathbf{r}_i]_2 & 2 \le j \le T+1\end{cases}\\
&\mathsf{sk}_Y = (\{\mathsf{sk}_{1,i}\}_{i\in[n]}, \{\mathsf{sk}_{2,i,j}\}_{i\in[n],j\in[T+1]}),
\end{aligned}$$

where $m'_{y,\ell,j}$ denotes the (ℓ, j)-th element of $\mathbf{M}'_{y_i}$.

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_X, \mathsf{sk}_Y)$**:** It takes pk, $\mathsf{ct}_X = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$, and $\mathsf{sk}_Y = (\{\mathsf{sk}_{1,i}\}_{i\in[n]}, \{\mathsf{sk}_{2,i,j}\}_{i\in[n],j\in[T+1]})$ such that $\mathsf{P}^{\mathsf{KP\text{-}MBF\text{-}16}}(X, Y) = 1$. Let S be the same as that defined in Pair of $\Gamma^{\mathsf{KP\text{-}MBF\text{-}16}}$. It outputs M' as follows.

$$M' = \mathsf{ct}_3 \Big/ \prod_{i\in S}\left(\left(\prod_{j\in[T+1]} e(\mathsf{ct}_1, \mathsf{sk}_{2,i,j})^{v''_{X,j}}\right) / e(\mathsf{ct}_2, \mathsf{sk}_{1,i})\right),$$

where $v''_{X,j}$ is the j-th element of $\mathbf{v}''_X$.