# Unbounded Dynamic Predicate Compositions in Attribute-Based Encryption

Nuttapong Attrapadung

National Institute of Advanced Industrial Science and Technology (AIST),
Tokyo, Japan.
n.attrapadung@aist.go.jp

**Abstract.** We present several transformations that combine a set of attribute-based encryption (ABE) schemes for simpler predicates into a new ABE scheme for more expressive composed predicates. Previous proposals for predicate compositions of this kind, the most recent one being that of Ambrona *et al.* at Crypto'17, can be considered *static* (or partially dynamic), meaning that the policy (or its structure) that specifies a composition must be fixed at the setup. Contrastingly, our transformations are *dynamic* and *unbounded*: they allow a user to specify an arbitrary and unbounded-size composition policy right into his/her own key or ciphertext. We propose transformations for three classes of composition policies, namely, the classes of any monotone span programs, any branching programs, and any deterministic finite automata. These generalized policies are defined over arbitrary predicates, hence admitting *modular* compositions. One application from modularity is a new kind of ABE for which policies can be "nested" over ciphertext and key policies. As another application, we achieve the first fully secure completely unbounded key-policy ABE for non-monotone span programs, in a modular and clean manner, under the q-ratio assumption. Our transformations work inside a generic framework for ABE called symbolic pair encoding, proposed by Agrawal and Chase at Eurocrypt'17. At the core of our transformations, we observe and exploit an unbounded nature of the symbolic property so as to achieve unbounded-size policy compositions.

## 1 Introduction

Attribute-based encryption (ABE), introduced by Sahai and Waters [32], is a paradigm that generalizes traditional public key encryption. Instead of encrypting to a target recipient, a sender can specify in a more general way about who should be able to view the message. In ABE for predicate $P : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, a ciphertext encrypting message $M$ is associated with a ciphertext attribute, say, $y \in \mathcal{Y}$, while a secret key, issued by an authority, is associated with a key attribute, say, $x \in \mathcal{X}$, and the decryption will succeed if and only if $P(x, y) = 1$. From an application point of view, we can consider one kind of attributes as *policies*, and the other kind as inputs to policies. In this sense, we have two basic forms of ABE called key-policy (KP) and ciphertext-policy (CP), depending on which side has a policy associated to.

**Predicate Compositions.** A central theme to ABE has been to expand the expressiveness by constructing new ABE for more powerful predicates (*e.g.,* [21,12,28,29,30,20]). In this work, we continue this theme by focusing on how to construct ABE for *compositions* of predicates. We are interested in devising *transformations* that combine ABE schemes for based predicates to a new ABE scheme for their composed predicate. To motivate that this can be powerful in the first place, we introduce an example pritimive called Nested-policy ABE.

**Example: Nested-policy ABE.** As the name suggests, it allows a key policy and a ciphertext policy to be nested to each other. This might be best described by an example. Suppose there are three categories for attributes: Person, Place, Content. Attached to a key, we could have attribute sets/policies categorized to three categories, Person:{Trainee, Doctor}, Place:{Paris, Zip:75001}, Content:'(Kidney and Disease) or Emergency', with a "composition policy" such as 'Person or (Place and Content)', which plays the role of concluding the whole policy. A ciphertext could be associated to Person:'Senior and Doctor', Place:'Paris or London', Content:{Kidney, Disease, Cancer}. Now we argue that the above key can be used to decrypt the ciphertext since the attribute set for Place satisfies the corresponding policy in the ciphertext, while the policy for Content is satisfied by the corresponding attribute sets in the ciphertext, and the concluding policy (attached to the key) states that if both Place and Content categories are satisfied, then it can decrypt.

We can consider this as a *composition* of two CP-ABE sub-schemes for the first two categories and KP-ABE for the last category, while on the top of that, a KP-ABE scheme over the three categories is then applied. To the best of our knowledge, no ABE with nested-policy functionality has been proposed so far, and it is not clear in the first place how to construct even for specific policies.

**Our Design Goal.** We aim at constructing *unbounded*, *dynamic*, and *generic* transformations for predicate compositions. *Dynamicity* refers to the property that one can choose *any* composition policy (defined in some sufficiently large classes) when composing predicates. In the above example, this translates to the property that the concluding policy is not fixed-once-and-for-all, where, for instance, one might want to define it instead as '(Person and Content) or Place', when a key is issued. Moreover, we aim at *modular* compositions where we can recursively define policies over policies, over and over again. Furthermore, for highest flexibility, we focus on *unbounded* compositions, meaning that the sizes of composition policies and attribute sets are not a-priori bounded at the setup. *Generality* refers to that we can transform *any* ABE for *any* based predicates. This level of generality might be too ambitious, since this would imply an attempt to construct ABE from ID-based Encryption (IBE), of which no transformation is known. We thus confine our goal to within some well-defined ABE framework and/or a class of predicates. Towards this, we first confine our attention to ABE based on *bilinear groups*, which are now considerably efficient and have always been the main tool for constructing ABE since the original papers [32,21].

**Previous Work on Predicate Compositions.** We categorize as follows.

- **Static & Specific**. Dual-policy ABE (DP-ABE), introduced in [4], is the AND composition of KP-ABE and CP-ABE (both fixed for the Boolean formulae predicate). The fixed AND means that it is static. The underlying ABE schemes are also specific schemes, namely, those of [21,33].
- **Static & Small-class & Generic**. Attrapadung and Yamada [10] proposed a more general conversion that can combine ABE for any predicates that can be interpreted in the so-called *pair encoding* framework [5,6,1,2], but again, fixed for only the AND connector. A generic DUAL conversion, which swaps key and ciphertext attribute, was also proposed in [5,10]. All in all, only a small class of compositions were possible at this point.
- **Static/Partially-dynamic & Large-class & Generic**. Most recently, at Crypto'17, Ambrona, Barthe, and Schmidt [3] proposed general tranformations for DUAL, AND, OR, and NOT connectors, hence complete any Boolean formulation, and thus enable a large class of combinations. Their scheme is generic and can combine ABE for any predicates in the so-called *predicate encoding* framework [36,16]. However, their compositions are static ones, where such a composition policy has to be fixed at the setup. A more flexible combination (§2 of [3]), which we call *partially dynamic*, is also presented, where the *structure* of the boolean combination must be fixed.

**Our Contributions: Dynamic & Large-class & Generic.** We propose *unbounded*, *dynamic*, and *generic* transformations for predicate compositions that contain a large class of policies. They are generic in the sense that applicable ABE schemes can be any schemes within the generic framework of pair encoding, see below. These transformation convert ABE schemes for a set of "atomic" predicates $\mathcal{P} = \{P_1, \dots, P_k\}$ to an ABE scheme for what we call *policy-augmented predicate over* $\mathcal{P}$. Both key-policy and ciphertext-policy augmentations are possible. In the key-policy case, the dynamicity allows a key issuer to specify a policy over atomic predicates, like the concluding policy over three sub-schemes in the above nested example. In the ciphertext-policy case, it allows an encryptor to specify such a policy. Below, we focus on the key-policy variant for illustrating purpose.

We propose the following four composition transformations.

1. **Span Programs over Predicates**. In this class, we let a composition policy be dynamically defined as any *monotone span program* (MSP) [22] where each of their Boolean inputs comes from each evaluation of atomic predicate. This is illustrated in Fig. 1. A key attribute is a tuple $M = (\mathbf{A}, (i_1, x_1), \dots, (i_m, x_m))$ depicted on the left, where $\mathbf{A}$ is a span program (or, think of it as a boolean formula). A ciphertext attribute is a set $Y = \{(j_1, y_1), \dots, (j_t, y_t)\}$. The indexes $i_d$ and $j_h$ specify the index of predicates in $\mathcal{P}$, that is, $i_d, j_h \in [1, k]$. To evaluate $M$ on $Y$, we proceed as follows. First, we evaluate a "link" between node $(i_d, x_d)$ and node $(j_h, y_h)$ to `on` if $i_d = j_h =: i$ and $P_i(x_d, y_h) = 1$. Then, if one of the edges adjacent to the $d$-th node is `on`, then we input 1 as the $d$-th input to $\mathbf{A}$, and evaluate $\mathbf{A}$. Our transformation is unbounded, meaning that $m$ and $t$ can be arbitrary. Note that since span programs imply boolean formulae, we can think of it as boolean formula over atomic predicates.

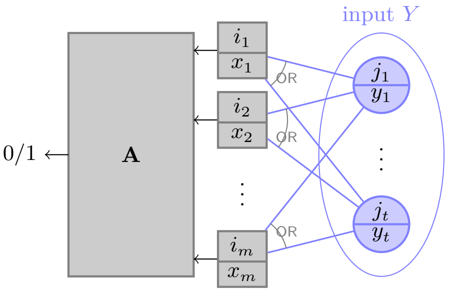


Fig. 1: Span program over predicates

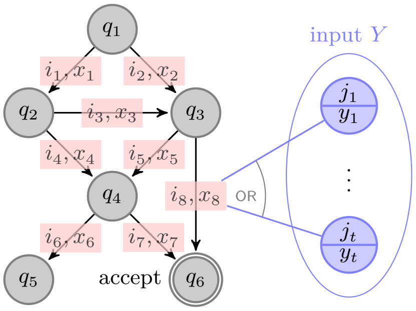


Fig. 2: Branching program over predicates

2. **Branching Programs over Predicates**. In this class, we let a composition policy be dynamically defined as any *branching program* (BP) where each edge is evaluated in a similar manner as in each link in the case of span program composition above. This is depicted in Fig. 2. A branching program is described by a direct acyclic graph (DAG) with labels. It accepts $Y$ if the `on` edges include a directed path from the start node to an accept node. A direct application for this is a predicate that comprises if-then clauses. We achieve this by a general implication from the first transformation, similarly to the implication from ABE for span programs to ABE for BP in [6].
3. **DFA over Predicates**. In this class, a composition policy can be defined as any *deterministic finite automata (DFA)* where each transition in DFA is defined based on atomic predicates. Such a DFA has an input as a vector $\mathbf{y} = ((j_1, y_1), \dots, (j_t, y_t))$ which it reads in sequence. It allows any direct graph, even contains directed cycles and loops (as opposed to DAG for branching programs), and can read arbitrarily long vectors $\mathbf{y}$. This transformation fully generalizes ABE for regular languages [35,5], which can deal only with the equality predicate at each transition, to any predicates.
4. **Bundling ABE with Parameter Reuse.** We propose a generic way to bundle ABE schemes (without a policy over them, and where each scheme works separately) so that almost all of their parameters can be set to the same set of values among those ABE schemes. This is quite surprising in the first place since usually parameters for different schemes would play different roles (in both syntax and security proof). Nevertheless, we show that they can be reused. Loosely speaking, to combine $k$ schemes where the maximum number of parameters (*i.e.,* public key size) among them is $n$, then the number of parameters for the combined scheme is $n+2k$. Trivially combining them would yield $O(nk)$ size. We call this as the *direct sum with parameter reuse.*

We denote the above first three key-policy-augmented predicates over $\mathcal{P}$ as $\mathsf{KP}[\mathcal{P}]$, $\mathsf{KB}[\mathcal{P}]$, $\mathsf{KA}[\mathcal{P}]$, respectively. For ciphertext-policy case, we use $\mathsf{C}$ instead of $\mathsf{K}$. Also, we call the generalized machines in the above classes as *predicative* machines.

**Scope of Our Transformations.** Our conversions apply to ABE that can be interpreted in the *pair encoding* framework, which is a generic framework

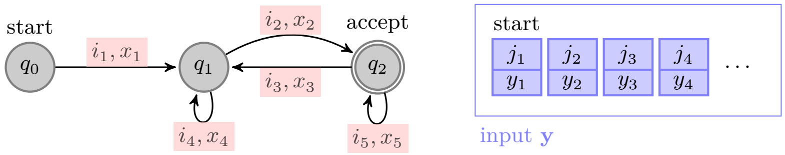


Fig. 3: DFA over predicates

for achieving fully secure ABE from a primitive called Pair Encoding Scheme (PES), proposed by Attrapadung [5]. PESs for many predicates have been proposed [5,10,6,2], notably, including regular language functionality [35,5]. Agrawal and Chase [2], at Eurocrypt'17, recently extended such a framework by introducing a notion called *symbolic security* for PES, which greatly simplifies both designing and security analysis of PES and ABE. A symbolically secure PES for predicate $P$ can be used to construct fully secure ABE for the same predicate under the $k$-linear and the q-ratio assumption [2] in (prime-order) bilinear groups. Our conversions indeed work by converting PESs for a set $\mathcal{P}$ of predicates to a PES for $\mathsf{KP}[\mathcal{P}]$, $\mathsf{KB}[\mathcal{P}]$, and $\mathsf{KA}[\mathcal{P}]$, that preserves symbolic security.

**Applications.** Among many applications, we obtain:

- ABE with multi-layer/multi-base functionalities and nested-policy. The generality of our transformations make it possible to augment ABE schemes in a *modular* and *recursive* manner. This enables multi-layer functionalities in one scheme, *e.g.,* ABE for predicate $\mathsf{KP}[\mathsf{KB}[\mathsf{KA}[\mathcal{P}]]]$, which can deal with first checking regular expression (over predicates) via DFA, then inputting to an if-clause in branching program, and finally checking the whole policy. By skewing key and ciphertext policy, we can obtain a nested-policy ABE, *e.g.,* predicate $\mathsf{KP}[\mathsf{CP}[\mathcal{P}]]$. Moreover, the fact that we combine a *set* of predicates into a composed one enables multiple based functionalities, *e.g.,* revocation [3,37], range/subset membership [8], regular string matching [35], etc. This level of "plug-and-play" was not possible before this work.
- The first fully secure *completely-unbounded* KP-ABE for *non-monotone* span programs (NSP) over large universe.[1] Previous ABE for NSP is either only selectively secure [28,9,38] or has some bounded attribute reuse [29,30]. See Table 1 in §9.2 for a summary. Our approach is simple as we can obtain this modularly. As a downside, we have to rely on the q-type assumption inherited from the Agrawal-Chase framework [2]. Nevertheless, all the current *completely unbounded* KP-ABE for even *monotone* span programs still need q-type assumptions [31,5,2], even selectively secure one [31].
- Mixed-policy ABE. In nested-policy ABE, the nesting structure is fixed. Mixed-policy ABE generalizes it so as to be able to deal with arbitrary nesting structure in one scheme. The scheme crucially uses the direct sum with parameter reuse, so that its parameter size will not blow up exponentially.

[1] For large-universe ABE, there is no known conversion from ABE for monotone span programs. Intuitively, one would have to include negative attributes for all of the complement of a considering attribute set, which is of exponential size.

**Comparing to** $\mathsf{ABS17}$ [3]. Here, we compare our transformations to those of Ambrona *et al.* [3]. The most distinguished features of our transformations are finite automata based, and branching program based compositions. Moreover, all of our transformations are unbounded. For monotone Boolean formulae over predicates, our framework allows dynamic compositions, as opposed to static or partially-dynamic (thus, bounded-size) ones in $\mathsf{ABS}$. As for applicability to based predicates, ours cover a larger class due to the different based frameworks (ours use symbolic pair encoding of [2], while $\mathsf{ABS}$ use predicate encoding of [16]). Notable differences are that pair encodings cover unbounded ABE for MSP, ABE for MSP with constant-size keys or ciphertexts, ABE for regular languages, while these are not known for predicate encodings. One drawback of using symbolic pair encoding is that we have to rely on q-type assumptions. A result in $\mathsf{ABS}$ also implies (static) *non-monotone* Boolean formulae composition (via their negation conversion). Although we do not consider negation conversion, we can use known pair encoding for negation of some common predicates such as IBE and negated of IBE (as we will do in §9). In this sense, non-monotone formulae composition can be done in our framework albeit in a semi-generic (but dynamic) manner.

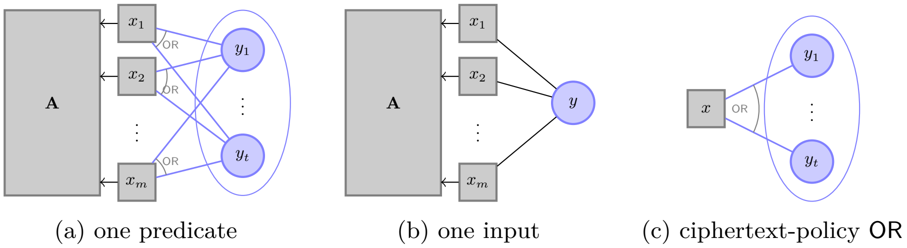
We provide more related works and some future directions in the full version.

## 2 Intuition and Informal Overview

This section provides some intuition on our approaches in an informal manner.

**Pair Encoding.** We first informally describe PES [5] as refined in [2]. It consists of two encoding algorithms as the main components. The ciphertext encoding $\mathsf{EncCt}$ encodes $y \in \mathcal{Y}$ to a vector $\mathbf{c} = \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) = (c_1, \ldots, c_{w_3})$ of polynomials in variables $\mathbf{s} = (s_0, \ldots, s_{w_1})$, $\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_{w_2})$, and $\mathbf{b} = (b_1, \ldots, b_n)$. The key encoding $\mathsf{EncKey}$ encodes $x \in \mathcal{X}$ to a vector $\mathbf{k} = \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) = (k_1, \ldots, k_{m_3})$ of polynomials in variables $\mathbf{r} = (r_1, \ldots, r_{m_1})$, $\hat{\mathbf{r}} = (\alpha, \hat{r}_1, \ldots, \hat{r}_{m_2})$, and $\mathbf{b}$. The correctness requires that if $P(x, y) = 1$, then we can "pair" $\mathbf{c}$ and $\mathbf{k}$ to to obtain $\alpha s_0$, which refers to the property that there exists a linear combination of terms $c_i r_u$ and $k_j s_t$ that is $\alpha s_0$. Loosely speaking, to construct ABE from PES, we use a bilinear group $\mathbb{G} = (\mathbb{G}_1, \mathbb{G}_2)$ that conforms to dual system groups [17,1,2]. Let $g_1, g_2$ be their generators. The public key is $(g_2^{\mathbf{b}}, e(g_1, g_2)^{\alpha})$, a ciphertext for $y$ encrypting a message $M$ consists of $g_2^{\mathbf{c}}, g_2^{\mathbf{s}}$, and $e(g_1, g_2)^{\alpha s_0} \cdot M$, and a key for $x$ consists of $g_1^{\mathbf{k}}, g_1^{\mathbf{r}}$. (In particular, the hatted variables are only internal to each encoding.) Decryption is done by pairing $\mathbf{c}$ and $\mathbf{k}$ to obtain $\alpha s_0$ in the exponent.

**Symbolic Security.** In a nutshell, the symbolic security [2] of PES involves "substitution" of scalar variables in PES to vectors/matrices so that all the substituted polynomials in the two encodings $\mathbf{c}$ and $\mathbf{k}$ will evaluate to zero for any pair $x, y$ such that $P(x, y) = 0$. The intuition for zero evaluation is that, behind the scene, there are some cancellations going on over values which cannot be computed from the underlying assumptions. To rule out the trivial all-zero substitutions, there is one more rule that the inner product of the substituted vectors for special variables that define correctness, namely, $\alpha$ and $s_0$, cannot



(a) one predicate (b) one input (c) ciphertext-policy OR

Fig. 4: Simpler variants of span program over predicates, for modular approach

be zero. In some sense, this can be considered as a generalization of the already well-known Boneh-Boyen cancellation technique for IBE [13].

Note that one has to prove two flavors of symbolic security: *selective* and *co-selective*. The former allows the substitutions of variables in $\mathbf{b}, \mathbf{c}$ to depend only on $y$, while those in $\mathbf{k}$ to depend on both $x, y$. In the latter, those in $\mathbf{b}, \mathbf{k}$ can depend only on $x$, while those in $\mathbf{c}$ can depend on both $x, y$. Intuitively, the framework of [2] uses each flavor in the two different phases—pre and post challenge—in the dual system proof methodology [34,23,26,36,5,2].

**Our Modular Approach.** In constructing a PES for $\mathsf{KP}[\mathcal{P}]$, we first look into the predicate definition itself and decompose to simpler ones as follows. Instead of dealing with predicates in the set $\mathcal{P}$ all at once, we consider its "direct sum", which allows us to view $\mathcal{P}$ as a single predicate, say $P$. Intuitively, this reduces $\mathsf{KP}[\mathcal{P}]$ of Fig. 1 to $\mathsf{KP}[P]$ of Fig. 4a. We then observe that $\mathsf{KP}[P]$ of Fig. 4a is, in fact, already a *nested* predicate. It contains ciphertext-policy with the OR policy in the lower layer, followed by key-policy augmentation in the upper layer, as decomposed and shown in Fig. 4c and Fig. 4b, respectively. Hence, we can consider a much simpler variant that deal with only one input at a time.

**Our Starting Point: Agrawal-Chase Unbounded ABE.** To illustrate the above decomposition, we consider a concrete predicate, namely, unbounded KP-ABE for monotone span program (MSP), along with a concrete PES, namely, an instantiation by Agrawal and Chase [2], which is, in fact, our starting point towards generalization. First we recall this PES (Appendix B.2 of [2])[2]:

$$\begin{aligned} \mathbf{c}_Y &= \big(b_1 s_0 + (y_j b_2 + b_3) s_1^{(j)}\big)_{j\in[q]} \\ \mathbf{k}_{(\mathbf{A},\pi)} &= \big(\mathbf{A}_i \hat{\mathbf{r}}^\top + r_1^{(i)} b_1,\, r_1^{(i)}(\pi(i) b_2 + b_3)\big)_{i\in[m]} \end{aligned} \tag{1}$$

where $(\mathbf{A}, \pi)$ is an MSP with $\mathbf{A} \in \mathbb{Z}_N^{m\times\ell}$, $\mathbf{A}_i$ is its $i$-th row, $\hat{\mathbf{r}} = (\alpha, \hat{r}_1, \ldots, \hat{r}_{\ell-1})$, and $Y = \{y_1, \ldots, y_q\}$. (The exact definition for MSP is not important for now.) We now attempt to view this as being achieved by two consecutive transformations.

[2] This encoding or closed variants are utilized in many works, *e.g.,* [25,31,5,18]. Rouselakis and Waters [31] were the first to (implicitly) use this exact encoding. Attrapadung [5] formalized it as PES. Agrawal and Chase [2] gave its symbolic proof.

We view the starting PES as the following PES for IBE ($P^{\mathsf{IBE}}(x, y) = 1$ iff $x = y$):

$$\begin{aligned} \mathbf{c}_y &= b_1 s_0 + (y b_2 + b_3) s_1 \\ \mathbf{k}_x &= \big(\alpha + r_1 b_1,\ r_1(x b_2 + b_3)\big) \end{aligned} \tag{2}$$

denoted as $\mathit{\Gamma}_{\mathsf{IBE}}$, which is first transformed to the following PES for IBBE (ID-based broadcast encryption, $P^{\mathsf{IBBE}}(x, Y) = 1$ iff $x \in Y$), denoted as $\mathit{\Gamma}_{\mathsf{IBBE}}$:

$$\begin{aligned} \mathbf{c}_Y &= \big(b_1 s_0 + (y_j b_2 + b_3) s_1^{(j)}\big)_{j\in[q]} = (c_j)_{j\in[q]} \\ \mathbf{k}_x &= \big(\alpha + r_1 b_1,\ r_1(x b_2 + b_3)\big) \end{aligned} \tag{3}$$

which is then finally transformed to the above PES for KP-ABE. We aim to generalize this process to any PES for arbitrary predicate.

The two transformations already comprise a nested policy augmentation process: the first (IBE to IBBE) is a ciphertext-policy one with the policy being simply the OR policy, while the second (IBBE to KP-ABE for MSP) is a key-policy one with policy $(\mathbf{A}, \pi)$. To see an intuition on a policy augmentation, we choose to focus on the first one here which is simpler since it is the OR policy. To see the relation of both PESs, we look into their matrix/vector substitutions in showing symbolic property. We focus on selective symbolic property here. It can be argued by showing matrix/vector substitutions that cause zero evaluations in all encodings, when $x \neq y$. For the base PES $\mathit{\Gamma}_{\mathsf{IBE}}$, this is:[3]

$$\begin{aligned} &\mathbf{c}_y : \overbrace{\boxed{\begin{smallmatrix}1\\0\end{smallmatrix}}}^{\mathbf{B}_1}\ \overset{(\mathbf{s}_0)^\top}{\overset{\uparrow}{1}} + \Big(y\overbrace{\boxed{\begin{smallmatrix}0\\-1\end{smallmatrix}}}^{\mathbf{B}_2} + \overbrace{\boxed{\begin{smallmatrix}-1\\y\end{smallmatrix}}}^{\mathbf{B}_3}\Big)\ \overset{(\mathbf{s}_1)^\top}{\overset{\uparrow}{1}} = \boxed{\begin{smallmatrix}0\\0\end{smallmatrix}} \\ &\mathbf{k}_x : \Big(1 + \boxed{-1,\ -\tfrac{1}{y-x}}\boxed{\begin{smallmatrix}1\\0\end{smallmatrix}} = 0,\ \ \boxed{-1,\ -\tfrac{1}{y-x}}\Big(x\boxed{\begin{smallmatrix}0\\-1\end{smallmatrix}} + \boxed{\begin{smallmatrix}-1\\y\end{smallmatrix}}\Big) = 0\Big) \end{aligned} \tag{4}$$

where each rectangle box represents a matrix of size 1 × 2 or 2 × 1. On the other hand, the selective symbolic property for the PES $\mathit{\Gamma}_{\mathsf{IBBE}}$ can be shown below, where we let $\mathbf{1}_j$ be the length-$q$ row vector with 1 at the $j$-th entry and $\mathbf{1}_{1,1}$ be the $(q+1) \times q$ matrix with 1 at the entry $(1, 1)$ (and all the other entries are 0).

$$\begin{aligned} &\mathbf{c}_Y : \overset{\mathbf{B}'_1}{\overset{\uparrow}{\mathbf{1}_{1,1}}}\overset{(\mathbf{s}'_0)^\top}{\overset{\uparrow}{(\mathbf{1}_1)^\top}} + \left(y_j \overbrace{\begin{pmatrix} 0 & \cdots & 0 \\ -1 & & \\ & \ddots & \\ & & -1 \end{pmatrix}}^{\mathbf{B}'_2} + \overbrace{\begin{pmatrix} -1 & \cdots & -1 \\ y_1 & & \\ & \ddots & \\ & & y_q \end{pmatrix}}^{\mathbf{B}'_3}\right)\overset{(\mathbf{s}'^{(j)}_1)^\top}{\overset{\uparrow}{(\mathbf{1}_j)^\top}} = 0 \\ &\mathbf{k}_x : \mathbf{1}_1 + \left(-1,\ -\tfrac{1}{y_1-x},\ \ldots,\ -\tfrac{1}{y_q-x}\right)\mathbf{1}_{1,1} = 0, \\ &\qquad \left(-1,\ -\tfrac{1}{y_1-x},\ \ldots,\ -\tfrac{1}{y_q-x}\right)\left(x\begin{pmatrix} 0 & \cdots & 0 \\ -1 & & \\ & \ddots & \\ & & -1 \end{pmatrix} + \begin{pmatrix} -1 & \cdots & -1 \\ y_1 & & \\ & \ddots & \\ & & y_q \end{pmatrix}\right) = 0. \end{aligned} \tag{5}$$

**Our Observation on Unboundedness.** We now examine the relation of substituted matrices/vectors between the two PESs: we observe that those for

[3] As a convention throughout the paper, the substitution matrices/vectors are written in the exact order of appearance in their corresponding encodings (here is Eq. (3)).

$\mathit{\Gamma}_{\mathsf{IBBE}}$ contains those for $\mathit{\Gamma}_{\mathsf{IBE}}$ as sub-matrices/vectors. For example, $\mathbf{B}_3$ for the substituted $\mathbf{c}_y$ in Eq. (4) is "embedded" in $\mathbf{B}'_3$ for the substituted $\mathbf{c}_Y$ in Eq. (5), for $y \in Y$. We denote such a sub-matrix as $\mathbf{B}_3^{(j)} = \left(\begin{smallmatrix} -1 \\ y_j \end{smallmatrix}\right)$.

We crucially observe that the unbounded property (of IBBE) stems from such an ability of embedding all the matrices from the base PES—$(\mathbf{B}_3^{(j)})_{j\in[q]}$—*regardless of size* $q$, into the corresponding matrix in the converted PES—$\mathbf{B}'_3$ in this case. Our aim is unbounded-size policy augmentation for *any* PES. We thus attempt to generalize this embedding process to work for any sub-matrices.

**Difficulty in Generalizing to Any PES.** Towards generalization, we could hope that such an embedding of sub-matrices/vectors has some patterns to follow. However, after a quick thought, we realize that the embedding here is quite specialized in many ways. The most obvious specialized form is the way that sub-matrices $\mathbf{B}_3^{(j)}$ are placed in $\mathbf{B}'_3$: the first row of $\mathbf{B}_3^{(j)}$ are placed in the same row in $\mathbf{B}'_3$, while the other row are placed in all different rows in $\mathbf{B}'_3$. Now the question is that such a special placement of sub-matrices into the composed matrices also applies to *any* generic PES. An answer for now is that this seems unlikely, if we do not restrict any structure of PES at all (which is what we aim).

We remark that, on the other hand, such a special embedding seems essential in our example here since, in each $c_j$, in order to cancel out the substitution of $b_1 s_0$, which is the same for all $j$, we must have the substitution for $(y_j b_2 + b_3)s_1^{(j)}$ to be the same for all $j \in [q]$. Therefore, we somehow must have a "projection" mechanism; this is enabled exactly by the placement in the first row of $\mathbf{B}'_2, \mathbf{B}'_3$.

**Our First Approach: Layering.** Our first approach is to modify the transformed PES so that sub-matrices can be placed in a "generic" manner into the composed matrices. (It will become clear shortly what we mean by "generic".) In the context of IBBE, we consider the following modified PES, denoted as $\bar{\mathit{\Gamma}}_{\mathsf{IBBE}}$:

$$\begin{aligned} \mathbf{c}_Y &= \big(f_2 s_{\mathrm{new}} + f_1 s_0^{(j)},\ b_1 s_0^{(j)} + (y_j b_2 + b_3)s_1^{(j)}\big)_{j\in[q]} \\ \mathbf{k}_x &= \big(\alpha_{\mathrm{new}} + r_{\mathrm{new}} f_2,\, r_{\mathrm{new}} f_1 + r_1 b_1,\, r_1(x b_2 + b_3)\big) \end{aligned} \qquad (6)$$

This is modified from the PES in Eq. (3) by introducing one more layer involving the first element in each encoding, where $f_1, f_2$ are two new parameters. The main purpose is to modify the element $b_1 s_0$ to $b_1 s_0^{(j)}$ so that it varies with $j$, which, in turn, eliminating the need for "projection" as previously. This becomes clear in the following assessment for its selective symbolic property:

$$
\begin{aligned}
&\mathbf{c}_Y : \hat{\mathbf{1}}_{1,1}(\mathbf{1}_1)^\top + \mathbf{F}_1(\mathbf{1}_j)^\top = 0,\\
&\quad \begin{pmatrix} \boxed{\begin{smallmatrix}1\\0\end{smallmatrix}} & & \\ & \ddots & \\ & & \boxed{\begin{smallmatrix}1\\0\end{smallmatrix}} \end{pmatrix} (\mathbf{1}_j)^\top + \left( y_j \begin{pmatrix} \boxed{\begin{smallmatrix}0\\-1\end{smallmatrix}} & & \\ & \ddots & \\ & & \boxed{\begin{smallmatrix}0\\-1\end{smallmatrix}} \end{pmatrix} + \begin{pmatrix} \boxed{\begin{smallmatrix}-1\\y_1\end{smallmatrix}} & & \\ & \ddots & \\ & & \boxed{\begin{smallmatrix}-1\\y_q\end{smallmatrix}} \end{pmatrix} \right) (\mathbf{1}_j)^\top = 0\\
&\mathbf{k}_x : \mathbf{1}_1 + (-\hat{\mathbf{1}}_1)\hat{\mathbf{1}}_{1,1} = 0,\\
&\qquad (-\hat{\mathbf{1}}_1)\mathbf{F}_1 + \left( \boxed{-1,\, -\tfrac{1}{y_1 - x}}, \ldots, \boxed{-1,\, -\tfrac{1}{y_q - x}} \right) \begin{pmatrix} \boxed{\begin{smallmatrix}1\\0\end{smallmatrix}} & & \\ & \ddots & \\ & & \boxed{\begin{smallmatrix}1\\0\end{smallmatrix}} \end{pmatrix} = 0,\\
&\qquad \left( \boxed{-1,\, -\tfrac{1}{y_1 - x}}, \ldots, \boxed{-1,\, -\tfrac{1}{y_q - x}} \right) \left( x \begin{pmatrix} \boxed{\begin{smallmatrix}0\\-1\end{smallmatrix}} & & \\ & \ddots & \\ & & \boxed{\begin{smallmatrix}0\\-1\end{smallmatrix}} \end{pmatrix} + \begin{pmatrix} \boxed{\begin{smallmatrix}-1\\y_1\end{smallmatrix}} & & \\ & \ddots & \\ & & \boxed{\begin{smallmatrix}-1\\y_q\end{smallmatrix}} \end{pmatrix} \right) = 0.
\end{aligned}
\tag{7}
$$

where we let $\hat{\mathbf{1}}_{1,1}$ be of size $(2q) \times q$ and $\hat{\mathbf{1}}_1$ be of length $2q$ (defined similarly to $\mathbf{1}_{1,1}, \mathbf{1}_1$, resp.), and let $\mathbf{F}_1$ be the $(2q) \times q$ matrix with all entries in the first row being $-1$ (and all the other entries are 0). Here, we observe that all the composed matrices regarding the parameters $(b_1, b_2, b_3)$ of the PES $\mathit{\Gamma}_{\mathsf{IBBE}}$ are formed exactly by including the substituted matrices of the base PES in the "diagonal blocks", namely, we can now "generically" define, for $i \in [n]$,

$$
\mathbf{B}'_i = \begin{pmatrix} \mathbf{B}_i^{(1)} & & \\ & \ddots & \\ & & \mathbf{B}_i^{(q)} \end{pmatrix}.
$$

Moreover, arranging the vector substitutions in their corresponding slots will result in exactly the zero evaluation of each substituted equation of the base PES. This approach is naturally generalized to any base PES. Put in other words, intuitively, we can obtain the proof of symbolic property of the composed PES from that of the base PES generically, via this conversion. Such a conversion, transforming any PES $(\mathbf{c}_y, \mathbf{k}_x)$ for predicate $P$ to its ciphertext-policy augmentation (with OR policy), can be described by

$$
\mathbf{c}'_Y = \left( f_2 s_{\mathrm{new}} + f_1 s_0^{(i)},\, \mathbf{c}_{y_j} \right)_{j \in [q]}, \quad \mathbf{k}'_x = \left( \alpha_{\mathrm{new}} + r_{\mathrm{new}} f_2,\, (\mathbf{k}_x)|_{\alpha \mapsto r_{\mathrm{new}} f_1} \right) \tag{8}
$$

where the variables $s_u$ in $\mathbf{c}_{y_j}$ are superscripted as $s_u^{(j)}$, and "$\mapsto$" denotes the variable replacement. This PES is for the predicate of "ciphertext-OR-policy" over $P$—returning true iff $\exists j\, P(x, y_j) = 1$. In fact, one can observe that Eq. (8) is a generalization of Eq. (6).

**Our Second Approach: Admissible PES.** One disadvantage with our first approach is the inefficiency due to the additional terms. Comparing PES $\bar{\mathit{\Gamma}}_{\mathsf{IBBE}}$

to $\mathit{\Gamma}_{\mathsf{IBBE}}$, the former requires $2q$ more elements than the latter (note that we include also $(s_0^{(j)})_{j\in[q]}$ when counting overall ciphertext elements). However, we already knew that the additional terms are not necessary for some specific PESs and predicates, notably our $\mathit{\Gamma}_{\mathsf{IBE}}$ for IBE.

We thus turn to the second approach which takes the following two steps. First, we find a class of "admissible" PESs where there exists a conversion for ciphertext-policy augmentation without additional terms. Second, we provide a conversion from any PES to a PES that is admissible.

As a result of our finding, the admissible class of PESs turns out to have a simple structure: $\mathbf{k}$ consists of $k_1 = \alpha + r_1 b_1$, and $\alpha, b_1$ do not appear elsewhere in $\mathbf{k}$, while in $\mathbf{c}$, we allow $b_1, s_0$ only if they are multiplied—$b_1 s_0$. Intuitively, this "isolation" of $b_1, \alpha, s_0$ somewhat provides a sufficient structure[4] where the "projection" can be enabled, but without mitigating to additional elements as done in the above first approach. The ciphertext-OR-policy augmentation can then be done by simply setting

$$\mathbf{c}'_Y = \big( (\mathbf{c}_{y_j})|_{s_0^{(j)} \mapsto s_{\mathrm{new}}} \big)_{j\in[q]}, \qquad\qquad \mathbf{k}'_x = \mathbf{k}_x. \tag{9}$$

One can observe that this is a generalization of Eq. (3), and that there is no additional terms as in Eq. (8). Our conversion from any PES to an admissible one (for the same predicate) is also simple: we set

$$\mathbf{c}'_y = \Big( f_2 s_{\mathrm{new}} + f_1 s_0,\ \ \mathbf{c}_y \Big), \qquad \mathbf{k}'_x = \Big( \alpha_{\mathrm{new}} + r_{\mathrm{new}} f_2,\ \ (\mathbf{k}_x)|_{\alpha \mapsto r_{\mathrm{new}} f_1} \Big) \tag{10}$$

where $s_0$ is the variable in $\mathbf{y}$, while $s_{\mathrm{new}}$ is the new special variable (that defines correctness). It is easy to see also that combining both conversions, that is, Eq. (10) followed by Eq. (9), we obtain the conversion of the first approach (Eq. (8)). But now, for any PES that is already admissible such as $\mathit{\Gamma}_{\mathsf{IBE}}$, we do not have to apply the conversion of Eq. (10), which requires additional terms.

**Towards General Policies.** Up to now, we only consider the OR policy. It ensures that $P'(x, Y) = 0$ implies $P(x, y_j) = 0$ for all $j$. However, for general policies, this is not the case, that is, if we let $\bar{P}$ be such a ciphertext-policy augmented predicate over $P$ (this will be formally given in Definition 5), $\bar{P}(x, (\mathbf{A}, \pi)) = 0$ may hold even if $P(x, \pi(j)) = 1$ for some $j$. Consequently, we have no available substituted matrices/vectors for the key encoding for such problematic $j$. Another important issue is how to embed the policy $(\mathbf{A}, \pi)$ without knowledge of $x$ (*cf.* the selective property), but be able to deal with any $x$ such that $\bar{P}(x, (\mathbf{A}, \pi)) = 0$.

We solve both simultaneously by a novel way of embedding $(\mathbf{A}, \pi)$ so that, intuitively, only the "non-problematic" blocks will turn "on", whatever $x$ will be, together with a novel way of defining substituted vectors for $\mathbf{k}$ so that all the "problematic" blocks will turn "off". To be able to deal with any $x$, the former has to be done in the "projection" part, while the latter is done in the

[4] Note that we indeed require a few more simple requirements in order for the proof to go through: see Definition 4.

"non-projection" part of matrices. By combining both, we will have only the non-problematic blocks turned on, and thus can use the base symbolic property.

**Towards Other Predicative Machines: Automata.** At the core of the above mechanism is the existence of "mask" vectors which render problematic blocks to 0. We crucially observe that such "mask" vectors depend on *and only on* $(x, (\mathbf{A}, \pi))$ and the sole fact that $\bar{P}(x, (\mathbf{A}, \pi)) = 0$, *i.e.,* the non-acceptance condition of MSP. Notably, it does not depend on the actual PES construction. This feature provides an insight to extend our approach to other types of predicative machines—finite automata in particular—by finding appropriate combinatorial vectors that encode non-acceptance conditions. (See more discussions in the full version.)

**Wrapping up.** Up to now, we mainly consider the selective symbolic property. The co-selective property (for the ciphertext-policy case) is simpler to achieve, since each substitution matrix of the converted PES is now required to embed only one matrix from the base PES, as our modular approach allows to consider one input at a time (for key attribute). The situation becomes reversed for the key-policy case: the co-selective property is harder. Nonetheless, we can always use the DUAL conversion to convert from ciphertext-policy to key-policy type.

**Comparing to Unboundedness Approach in** CGKW [18]. Chen *et al.* [18] recently proposed unbounded ABE for MSP. Their approach conceptually converts a specific bounded scheme ([27]) to an unbounded one for the *same* specific predicate—MSP. This is already semantically different to our conversion, which takes any pair encoding for a predicate $P$ and outputs another for a *different* predicate—namely, the (unbounded) policy-augmented predicate over $P$.

## 3 Preliminaries

**Notations.** $\mathbb{N}$ denotes the set of positive integers. For $a, b \in \mathbb{N}$ such that $a \leq b$, let $[a, b] = \{ a, \dots, b \}$. For $m \in \mathbb{N}$, let $[m] = \{ 1, \dots, m \}$ and $[m]^+ = \{ 0, 1, \dots, m \}$. For a set $S$, we denote by $2^S$ the set of all subsets of $S$. Denote by $S^*$ the set of all (unbounded-length) sequences where each element is in $S$. For $N \in \mathbb{N}$, we denote by $\mathbb{Z}_N^{m \times \ell}$ the set of all matrices of dimension $m \times \ell$ with elements in $\mathbb{Z}_N$. For a matrix $\mathbf{M} \in \mathbb{Z}_N^{m \times \ell}$, its $i$-th row vector is denoted by $\mathbf{M}_{i:}$ (in $\mathbb{Z}_N^{1 \times \ell}$). Its $(i, j)$-element is $\mathbf{M}_{i,j}$. Its transpose is denoted as $\mathbf{M}^\top$. For vectors $\mathbf{a} \in \mathbb{Z}_N^{1 \times c}$,$\mathbf{b} \in \mathbb{Z}_N^{1 \times d}$, we denote $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_N^{1 \times (c+d)}$ as the concatenation. The $i$-th entry of $\mathbf{a}$ is denoted as $\mathbf{a}[i]$. For $i < j$, denote $\mathbf{a}[i, j] \coloneqq (\mathbf{a}[i], \mathbf{a}[i+1], \dots, \mathbf{a}[j])$. Let $\mathbb{M}(\mathbb{Z}_N)$ be the set of all matrices (of any sizes) in $\mathbb{Z}_N$, and $\mathbb{M}_m(\mathbb{Z}_N)$ be the set of those with $m$ rows. For a set $S$ of vectors of the same length (say, in $\mathbb{Z}_N^\ell$), we denote $\mathrm{span}(S)$ as the set of all linear combinations of vectors in $S$. For polynomials $p = p(x_1, \dots, x_n)$ and $g = g(y_1, \dots, y_n)$, we denote a new polynomial $p|_{x_1 \mapsto g} \coloneqq p(g(y_1, \dots, y_n), x_2, \dots, x_n)$. Matrices and vectors with all 0's are simply denoted by 0, of which the dimension will be clear from the context. We define some useful fixed vectors and matrices.

- $\mathbf{1}_i^\ell$ is the (row) vector of length $\ell$ with 1 at position $i$ where all others are 0.
- $\mathbf{1}_{i,j}^{m \times \ell}$ is the matrix of size $m \times \ell$ with 1 at position $(i, j)$ and all others are 0.

### 3.1 Definitions for General ABE

**Predicate Family.** Let $P = \{\, P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\} \mid \kappa \in \mathcal{K} \,\}$ be a predicate family where $\mathcal{X}_\kappa$ and $\mathcal{Y}_\kappa$ denote "key attribute" and "ciphertext attribute" spaces. The index $\kappa$ or "parameter" denotes a list of some parameters such as the universes of attributes, and/or bounds on some quantities, hence its domain $\mathcal{K}$ will depend on that predicate. We will often omit $\kappa$ when the context is clear.

**General ABE Syntax.** Let $\mathcal{M}$ be a message space. An ABE scheme[5] for predicate family $P$ is defined by the following algorithms:

- $\mathsf{Setup}(1^\lambda, \kappa) \to (\mathsf{PK}, \mathsf{MSK})$: takes as input a security parameter $1^\lambda$ and a parameter $\kappa$ of predicate family $P$, and outputs a master public key $\mathsf{PK}$ and a master secret key $\mathsf{MSK}$.
- $\mathsf{Encrypt}(y, M, \mathsf{PK}) \to \mathsf{CT}$: takes as input a ciphertext attribute $y \in \mathcal{Y}_\kappa$, a message $M \in \mathcal{M}$, and public key $\mathsf{PK}$. It outputs a ciphertext $\mathsf{CT}$. We assume that $Y$ is implicit in $\mathsf{CT}$.
- $\mathsf{KeyGen}(x, \mathsf{MSK}, \mathsf{PK}) \to \mathsf{SK}$: takes as input a key attribute $x \in \mathcal{X}_\kappa$ and the master key $\mathsf{MSK}$. It outputs a secret key $\mathsf{SK}$.
- $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK}) \to M$: given a ciphertext $\mathsf{CT}$ with its attribute $y$ and the decryption key $\mathsf{SK}$ with its attribute $x$, it outputs a message $M$ or $\perp$.

**Correctness.** Consider all parameters $\kappa$, all $M \in \mathcal{M}$, $x \in \mathcal{X}_\kappa$, $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 1$. If $\mathsf{Encrypt}(y, M, \mathsf{PK}) \to \mathsf{CT}$ and $\mathsf{KeyGen}(x, \mathsf{MSK}, \mathsf{PK}) \to \mathsf{SK}$ where $(\mathsf{PK}, \mathsf{MSK})$ is generated from $\mathsf{Setup}(1^\lambda, \kappa)$, then $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK}) \to M$.

**Security.** We use the standard notion for ABE, called full security. We omit it here and refer to *e.g.,* [5] (or the full version of this paper), as we do not work directly on it but will rather infer the implication from pair encoding scheme (*cf.* §3.3).

**Duality of ABE.** For a predicate $P : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, we define its dual as $\bar{P} : \mathcal{Y} \times \mathcal{X} \to \{0,1\}$ by setting $\bar{P}(Y, X) = P(X, Y)$. In particular, if $P$ is considered as key-policy type, then its dual, $\bar{P}$, is the corresponding ciphertext-policy type.

### 3.2 Pair Encoding Scheme Definition

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

- $\mathsf{Param}(\mathsf{par}) \to n$. When given $\mathsf{par}$ as input, $\mathsf{Param}$ outputs $n \in \mathbb{N}$ that specifies the number of *common* variables, which we denote by $\mathbf{b} := (b_1, \ldots, b_n)$.

[5] It is also called public-index predicate encryption, classified in the definition of Functional Encryption [15]. It is simply called predicate encryption in [2].

- $\mathsf{EncCt}(y, N) \to (w_1, w_2, \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}))$. On input $N \in \mathbb{N}$ and $y \in \mathcal{Y}_{(N,\mathsf{par})}$, $\mathsf{EncCt}$ outputs a vector of polynomial $\mathbf{c} = (c_1, \ldots, c_{w_3})$ in *non-lone* variables $\mathbf{s} = (s_0, s_1, \ldots, s_{w_1})$ and *lone* variables $\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_{w_2})$. For $p \in [w_3]$, the $p$-th polynomial is given as follows, where $\eta_{p,z}, \eta_{p,t,j} \in \mathbb{Z}_N$:
$$\sum_{z \in [w_2]} \eta_{p,z} \hat{s}_z + \sum_{t \in [w_1]^+, j \in [n]} \eta_{p,t,j} b_j s_t.$$
- $\mathsf{EncKey}(x, N) \to (m_1, m_2, \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}))$. On input $N \in \mathbb{N}$ and $x \in \mathcal{X}_{(N,\mathsf{par})}$, $\mathsf{EncKey}$ 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 $\hat{\mathbf{r}} = (\alpha, \hat{r}_1, \ldots, \hat{r}_{m_2})$. For $p \in [m_3]$, the $p$-th polynomial is given as follows, where $\phi_p, \phi_{p,u}, \phi_{p,v,j} \in \mathbb{Z}_N$:
$$\phi_p \alpha + \sum_{u \in [m_2]} \phi_{p,u} \hat{r}_u + \sum_{v \in [m_1], j \in [n]} \phi_{p,v,j} r_v b_j.$$
- $\mathsf{Pair}(x, y, N) \to (\mathbf{E}, \overline{\mathbf{E}})$. On input $N$, and both $x$, and $y$, $\mathsf{Pair}$ outputs two matrices $\mathbf{E}, \overline{\mathbf{E}}$ of sizes $(w_1 + 1) \times m_3$ and $w_3 \times m_1$, respectively. $\Diamond$

**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 $P_\kappa(x, y) = 1$, the following holds symbolically:
$$\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \alpha s_0. \tag{11}$$
The left-hand side is indeed a linear combination of $s_t k_p$ and $c_q r_v$, for $t \in [w_1]^+, p \in [m_3], q \in [w_3], v \in [m_1]$. Hence, an equivalent (and somewhat simpler) way to describe $\mathsf{Pair}$ and correctness together at once is to show such a linear combination that evaluates to $\alpha s_0$. We will use this approach throughout the paper. (The matrices $\mathbf{E}, \overline{\mathbf{E}}$ will be implicitly defined in such a linear combination).

**Terminology.** In the above, following [2], a variable is called *lone* as it is not multiplied with any $b_j$ (otherwise called *non-lone*). Furthermore, since $\alpha$, $s_0$ are treated distinguishably in defining correctness, we also often call them the *special* lone and non-lone variable, respectively. In what follows, we use $\mathsf{ct\text{-}enc}$ and $\mathsf{key\text{-}enc}$ as a shorthand for polynomials and variables output by $\mathsf{EncCt}$ (ciphertext-encoding) and $\mathsf{EncKey}$ (key-encoding), respectively. We often omit writing $w_1, w_2$ and $m_1, m_2$ in the output of $\mathsf{EncCt}$ and $\mathsf{EncKey}$.

### 3.3 Symbolic Property of PES

We now describe the symbolic property of PES, introduced in [2]. As in [2], we use $a : b$ to denote that a variable $a$ is substituted by a matrix/vector $b$.

**Definition 2.** A PES $\Gamma = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ for predicate family $P$ satisfies $(d_1, d_2)$-*selective symbolic property* for some $d_1, d_2 \in \mathbb{N}$ if there exists three deterministic polynomial-time algorithms $\mathsf{EncB}, \mathsf{EncS}, \mathsf{EncR}$ such that for all $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$, $y \in \mathcal{Y}_\kappa$ with $P_\kappa(x, y) = 0$,

- $\mathsf{EncB}(y) \to \mathbf{B}_1, \ldots, \mathbf{B}_n \in \mathbb{Z}_N^{d_1 \times d_2}$;
- $\mathsf{EncS}(y) \to \mathbf{s}_0, \ldots, \mathbf{s}_{w_1} \in \mathbb{Z}_N^{1 \times d_2}, \quad \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2} \in \mathbb{Z}_N^{1 \times d_1}$;
- $\mathsf{EncR}(x, y) \to \mathbf{r}_1, \ldots, \mathbf{r}_{m_1} \in \mathbb{Z}_N^{1 \times d_1}, \quad \mathbf{a}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2} \in \mathbb{Z}_N^{1 \times d_2}$;

we have that:

(P1). $\mathbf{a}\mathbf{s}_0^\top \neq 0$.

(P2). if we substitute, for all $j \in [n]$, $t \in [w_1]^+$, $z \in [w_2]$, $v \in [m_1]$, $u \in [m_2]$,

$$\hat{s}_z : \hat{\mathbf{s}}_z^\top, \qquad b_j s_t : \mathbf{B}_j \mathbf{s}_t^\top, \qquad \alpha : \mathbf{a}, \qquad \hat{r}_u : \hat{\mathbf{r}}_u, \qquad r_v b_j : \mathbf{r}_v \mathbf{B}_j,$$

into all the polynomials output by $\mathsf{EncCt}(y)$ and $\mathsf{EncKey}(x)$, then they evaluate to 0.

(P3). $\mathbf{a} = \mathbf{1}_1^{d_2}$.

Similarly, a PES satisfies $(d_1, d_2)$-*co-selective symbolic property* if there exists $\mathsf{EncB}, \mathsf{EncS}, \mathsf{EncR}$ satisfying the above properties but where $\mathsf{EncB}$ and $\mathsf{EncR}$ depends only on $x$, and $\mathsf{EncS}$ depends on both $x$ and $y$.

Finally, a PES satisfies $(d_1, d_2)$-*symbolic property* if it satisfies both $(d_1', d_2')$-selective and $(d_1'', d_2'')$-co-selective properties for some $d_1', d_1'' \leq d_1$, $d_2', d_2'' \leq d_2$. $\diamondsuit$

**Terminology.** The original definition in [2] consists of only (P1) and (P2); we refer to this as Sym-Prop, as in [2]. We newly include (P3) here, and refer to the full definition with all (P1)-(P3) as Sym-Prop$^+$. This is w.l.o.g. since one can convert any PES with Sym-Prop to another with Sym-Prop$^+$, with minimal cost. Such a conversion, which we denote as Plus-Trans, also appears in [2]; we recap it in the full version.

For convenience, for the case of selective property, we use $\mathsf{EncBS}(y)$ to simply refer to the concatenation of $\mathsf{EncB}(y)$ and $\mathsf{EncS}(y)$. Similarly, we use $\mathsf{EncBR}(x)$ for referring $\mathsf{EncB}(x)$ and $\mathsf{EncR}(x)$ for the case of co-selective property.

**Implication to Fully Secure ABE.** Agrawal and Chase [2] show that a PES satisfying $(d_1, d_2)$-Sym-Prop implies fully secure ABE. They use an underlying assumption called $(D_1, D_2)$-q-ratio, which can be defined in the dual system groups [17] and can consequently be instantiated in the prime-order bilinear groups. Note that paramater $(D_1, D_2)$ are related to $(d_1, d_2)$. Since their theorem is not used explicitly in this paper, we recap it in the full version.

### 3.4 Definitions for Some Previous Predicates

**ABE for Monotone Span Program.** We recap the predicate definition for KP-ABE for monotone span program (MSP) [21]. We will mostly focus on *completely unbounded* variant [5,2], where the family index is simply $\kappa = N \in \mathbb{N}$, that is, any additional parameter par is not required.[6] Below, we also state a useful lemma which is implicit in *e.g.,* [21,27].

[6] Bounded schemes would use par for specifying some bounds, *e.g.,* on policy or attribute set sizes, or the number of attribute multi-use in one policy. The term "Unbounded ABE" used in the literature [25,30,18] still allows to have a bound for the number of attribute multi-use in one policy (or even a one-use restriction).

**Definition 3.** The predicate family of *completely unbounded KP-ABE for monotone span programs*, $P^{\textsf{KP-MSP}} = \{\, P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\} \,\}_\kappa$, is indexed by $\kappa = (N)$ and is defined as follows. Recall that $\mathbf{A}_{i:}$ denotes the $i$-th row of $\mathbf{A}$.

- $\mathcal{X}_\kappa = \{\, (\mathbf{A}, \pi) \mid \mathbf{A} \in \mathbb{M}(\mathbb{Z}_N),\ \pi : [m] \to \mathbb{Z}_N \,\}$.
- $\mathcal{Y}_\kappa = 2^{(\mathbb{Z}_N)}$.
- $P_\kappa((\mathbf{A}, \pi), Y) = 1 \iff \mathbf{1}_1^\ell \in \operatorname{span}(\mathbf{A}|_Y)$, where $\mathbf{A}|_Y := \{\, \mathbf{A}_{i:} \mid \pi(i) \in Y \,\}$.

where $m \times \ell$ is the size of the matrix $\mathbf{A}$. ♢

**Proposition 1.** *Consider a matrix $\mathbf{A} \in \mathbb{Z}_N^{m\times\ell}$. Let $Q \subseteq [m]$ be a set of row indexes. If $\mathbf{1}_1^\ell \notin \operatorname{span}\{\, \mathbf{A}_{i:} \mid i \in Q \,\}$, then there exists $\boldsymbol{\omega} = (w_1, \ldots, w_\ell) \in \mathbb{Z}_N^\ell$ such that $w_1 = 1$ and $\mathbf{A}_{i:}\boldsymbol{\omega}^\top = 0$ for all $i \in Q$.*

**Specific Policies.** It is well known that ABE for MSP implies ABE for monotone Boolean formulae [21,11]. The procedure of embedding a boolean formula as a span program can be found in *e.g.,* §C of [24]. We will be interested in the OR and the AND policy, for using as building blocks later on. For the OR policy, the access matrix is of the form $\mathbf{A}_{\mathsf{OR},m} = (1, \ldots, 1)^\top \in \mathbb{Z}_N^{m\times 1}$. For the AND policy, it is $\mathbf{A}_{\mathsf{AND},m} = \sum_{i=1} \mathbf{1}_{i,i}^{m\times m} - \sum_{j=2} \mathbf{1}_{1,j}^{m\times m}$. For further use, we let $\mathbb{M}_{\mathsf{OR}}(\mathbb{Z}_N) = \left\{\, \mathbf{A}_{\mathsf{OR},m} \,\middle|\, m \in \mathbb{N} \,\right\}$ and $\mathbb{M}_{\mathsf{AND}}(\mathbb{Z}_N) = \left\{\, \mathbf{A}_{\mathsf{AND},m} \,\middle|\, m \in \mathbb{N} \,\right\}$.

**Embedding Lemma.** To argue that a PES for predicate $P$ can be used to construct a PES for predicate $P'$, intuitively, it suffices to find mappings that map attributes in $P'$ to those in $P$, and argue that the predicate evaluation for $P'$ is preserved to that for $P$ on the mapped attributes. In such a case, we say that $P'$ *can be embedded into* $P$. This is known as the embedding lemma, used for general ABE in [14,7]. We prove the implication for the case of PES in the full version.

## 4 Admissible Pair Encodings

We first propose the notion of *admissible PES*. It is a class of PESs where a conversion to a new PES for its policy-augmented predicate exists without additional terms, as motivated in the second approach in §2. We then provide a conversion from *any* PES to an admissible PES of the same predicate (this, however, poses additional terms).[7] Together, these thus allow us to convert any PES to a new PES for its policy-augmented predicate.

**Definition 4.** A PES is $(d_1, d_2)$-admissible if it satisfies $(d_1, d_2)$-$\textsf{Sym-Prop}^+$ with the following additional constraints.

(P4). In the key encoding $\mathbf{k}$, the first polynomial has the form $k_1 = \alpha + r_1 b_1$ and $\alpha, b_1$ do not appear elsewhere in $\mathbf{k}$.

[7] Interestingly, this conversion already appears in [2] but for different purposes.

(P5). In the ciphertext encoding $\mathbf{c}$, the variables $b_1$ and $s_0$ can only appear in the term $b_1 s_0$.[8]

(P6). In the symbolic property (both selective and co-selective), we have that $\mathbf{B}_1 = \mathbf{1}_{1,1}^{d_1 \times d_2}$, $\mathbf{s}_0 = \mathbf{1}_1^{d_2}$, and $\mathbf{r}_v[1] \neq 0$ for all $v \in [m_1]$. $\Diamond$

We will use the following for the correctness of our conversion in §5.

**Corollary 1.** *For any admissible PES, let* $\mathbf{c}, \mathbf{k}, \mathbf{s}, \mathbf{r}, \mathbf{E}, \overline{\mathbf{E}}$ *be defined as in Definition 1 with* $P_\kappa(x, y) = 1$. *Let* $\tilde{\mathbf{s}} = (s_1, \ldots, s_{w_1})$. *There exists a PPT algorithm that takes* $\mathbf{E}$ *and outputs a matrix* $\tilde{\mathbf{E}}$ *of size* $w_1 \times m_3$ *such that* $\tilde{\mathbf{s}}\tilde{\mathbf{E}}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = -r_1 b_1 s_0$.

*Proof.* We re-write Eq. (11) as $s_0 k_1 + T + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \alpha s_0$ (where $T$ is a sum of $s_t k_j$ with coefficients from $\mathbf{E}$). Note that $s_0 k_1$ has coefficient 1 since $\alpha$ appears only in $k_1$ and we match the monomial $\alpha s_0$ to the right hand side. Substituting $k_1 = \alpha + r_1 b_1$, we have $T + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = -r_1 b_1 s_0$. We claim that $s_0$ is not in $T$, which would prove the corollary. To prove the claim, we first see that $k_1$ is not in $T$, since $\alpha$ is not in the right hand side. Thus $b_1$ is also not in $T$ (as $b_1$ only appears in $k_1$). Hence, $s_0$ is not in $T$, since otherwise $b_j s_0$ where $j \geq 2$ appears in $T$, but in such a case, it cannot be cancelled out since such term is not allowed in $\mathbf{c}$. □

**Construction 1.** Let $\Gamma$ be a PES construction for $P$. We construct another PES $\Gamma'$ for also the same $P$ as follows. We denote this $\Gamma'$ by $\mathsf{Layer\text{-}Trans}(\Gamma)$.

- $\mathsf{Param}'(\mathsf{par})$. If $\mathsf{Param}(\mathsf{par})$ returns $n$, then output $n+2$. Denote $\mathbf{b} = (b_1, \ldots, b_n)$ and $\mathbf{b}' = (f_1, f_2, \mathbf{b})$.
- $\mathsf{EncCt}'(y, N)$. Run $\mathsf{EncCt}(y, N) \to \mathbf{c}$. Let $s_0$ be the special variable in $\mathbf{c}$. Let $s_{\text{new}}$ be the new special variable. Output $\mathbf{c}' = (f_1 s_{\text{new}} + f_2 s_0,\ \mathbf{c})$.
- $\mathsf{EncKey}'(x, N)$. Run $\mathsf{EncKey}(x, N) \to \mathbf{k}$. Let $r_{\text{new}}$ be a new non-lone variable and $\alpha_{\text{new}}$ be the new special lone variable. Let $\tilde{\mathbf{k}}$ be exactly $\mathbf{k}$ but with $\alpha$ being replaced by $r_{\text{new}} f_2$. Output $(\alpha_{\text{new}} + r_{\text{new}} f_1,\ \tilde{\mathbf{k}})$.

**Pair/Correctness.** Suppose $P(x, y) = 1$. From the correctness of $\Gamma$ we have a linear combination that results in $\alpha s_0 = r_{\text{new}} f_2 s_0$. From then, we have $(\alpha_{\text{new}} + r_{\text{new}} f_1) s_{\text{new}} - r_{\text{new}}(f_1 s_{\text{new}} + f_2 s_0) + r_{\text{new}} f_2 s_0 = \alpha_{\text{new}} s_{\text{new}}$, as required.

**Lemma 1.** *Suppose that* $\Gamma$ *for* $P$ *satisfies* $(d_1, d_2)$-$\mathsf{Sym\text{-}Prop}^+$. *Then, the PES* $\mathsf{Layer\text{-}Trans}(\Gamma)$ *for* $P$ *is* $(d_1 + 1, d_2)$*-admissible.* (The proof is deferred to the full version.)

## 5 Ciphertext-policy Augmentation

We now describe the notion of ciphertext-policy-span-program-augmented predicate over a *single* predicate family. We then construct a conversion that preserves admissibility. The case for a *set* of predicate families will be described in §7. The key-policy case will be in the next section §6.

[8] That is, $b_j s_0$ and $b_1 s_t$ for $j \in [2, n], t \in [1, n]$ are not allowed in $\mathbf{c}$.

**Definition 5.** Let $P = \{ P_\kappa \}_\kappa$ where $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{ 0, 1 \}$, be a predicate family. We define the *ciphertext-policy-span-program-augmented predicate* over $P$ as $\mathsf{CP1}[P] = \{ \bar{P}_\kappa \}_\kappa$ where $\bar{P}_\kappa : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{ 0, 1 \}$ by letting

- $\bar{\mathcal{X}}_\kappa = \mathcal{X}_\kappa$.
- $\bar{\mathcal{Y}}_\kappa = \{ (\mathbf{A}, \pi) \mid \mathbf{A} \in \mathbb{M}(\mathbb{Z}_N),\ \pi : [m] \to \mathcal{Y}_\kappa \}$.
- $\bar{P}_\kappa(x, (\mathbf{A}, \pi)) = 1 \iff \mathbf{1}_1^\ell \in \mathrm{span}(\mathbf{A}|_x)$, where $\mathbf{A}|_x \coloneqq \{ \mathbf{A}_{i:} \mid P_\kappa(x, \pi(i)) = 1 \}$.

where $m \times \ell$ is the size of the matrix $\mathbf{A}$. $\diamondsuit$

**Construction 2.** Let $\mathit{\Gamma}$ be a PES construction for $P$ satisfying admissibility. We construct a PES $\mathit{\Gamma}'$ for $\mathsf{CP1}[P]$ as follows. Denote this $\mathit{\Gamma}'$ by $\mathsf{CP1\text{-}Trans}(\mathit{\Gamma})$.

- $\mathsf{Param}'(\mathsf{par}) = \mathsf{Param}(\mathsf{par}) = n$. Denote $\mathbf{b} = (b_1, \ldots, b_n)$.
- $\mathsf{EncKey}'(x, N) = \mathsf{EncKey}(x, N)$.
- $\mathsf{EncCt}'((\mathbf{A}, \pi), N)$. Parse $\mathbf{A} \in \mathbb{Z}_N^{m \times \ell}$.
  - For $i \in [m]$, run $\mathsf{EncCt}(\pi(i), N)$ to obtain a vector $\mathbf{c}^{(i)} = \mathbf{c}^{(i)}(\mathbf{s}^{(i)}, \hat{\mathbf{s}}^{(i)}, \mathbf{b})$ of polynomials in variables $\mathbf{s}^{(i)} = (s_0^{(i)}, s_1^{(i)}, \ldots, s_{w_{1,i}}^{(i)})$, $\hat{\mathbf{s}}^{(i)} = (\hat{s}_1^{(i)}, \ldots, \hat{s}_{w_{2,i}}^{(i)})$, and $\mathbf{b}$. Denote $\tilde{\mathbf{s}}^{(i)} = (s_1^{(i)}, \ldots, s_{w_{1,i}}^{(i)})$.
  - Let $s_{\mathrm{new}}$ be the new special non-lone variable. Let $v_2, \ldots, v_\ell$ be new lone variables. Denote $\mathbf{v} = (b_1 s_{\mathrm{new}}, v_2, \ldots, v_\ell)$.
  - For $i \in [m]$, define a modified vector by variable replacement as

$$\mathbf{c}'^{(i)} \coloneqq \mathbf{c}^{(i)}|_{b_1 s_0^{(i)} \mapsto \mathbf{A}_{i:} \mathbf{v}^\top}. \tag{12}$$

  Finally, output $\mathbf{c}' = \mathbf{c}'(\mathbf{s}', \hat{\mathbf{s}}', \mathbf{b}')$ as $\mathbf{c}' = \big(\mathbf{c}'^{(i)}\big)_{i \in [m]}$. It contains variables $\mathbf{s}' = \big(s_{\mathrm{new}}, \big(\tilde{\mathbf{s}}^{(i)}\big)_{i \in [m]}\big)$, $\hat{\mathbf{s}}' = \big(v_2, \ldots, v_\ell, \big(\,\hat{\mathbf{s}}^{(i)}\,\big)_{i \in [m]}\big)$, and $\mathbf{b}'$.

**Pair/Correctness.** For proving correctness, we suppose $\bar{P}_\kappa(x, (\mathbf{A}, \pi)) = 1$. Let $S \coloneqq \{ i \in [m] \mid P_\kappa(x, \pi(i)) = 1 \}$. For $i \in S$, we can run $\mathsf{Pair}(x, \pi(i), N) \to (\mathbf{E}, \overline{\mathbf{E}})$. From the correcteness of $\mathit{\Gamma}$, we derive $\tilde{\mathbf{E}}$ from $\mathbf{E}$ via Corollary 1, and obtain a linear combination $\tilde{\mathbf{s}}^{(i)} \tilde{\mathbf{E}} \mathbf{k}^\top + \mathbf{c}^{(i)} \overline{\mathbf{E}} \mathbf{r}^\top = -r_1 b_1 s_0^{(i)}$. With the variable replacement in Eq. (12), this becomes $\tilde{\mathbf{s}}^{(i)} \tilde{\mathbf{E}} \mathbf{k}^\top + \mathbf{c}'^{(i)} \overline{\mathbf{E}} \mathbf{r}^\top = -r_1 \mathbf{A}_{i:} \mathbf{v}^\top$. Now since $\mathbf{1}_1^\ell \in \mathrm{span}(\mathbf{A}|_x)$, we have linear combination coefficients $\{ t_i \}_{i \in S}$ such that $\sum_{i \in S} t_i \mathbf{A}_{i:} = \mathbf{1}_1^\ell$. Hence we have the following linear combination, as required:[9] $k_1 s_{\mathrm{new}} + \sum_{i \in S} t_i \big( - r_1 \mathbf{A}_{i:} \mathbf{v}^\top \big) = (\alpha + r_1 b_1) s_{\mathrm{new}} - r_1 b_1 s_{\mathrm{new}} = \alpha_{\mathrm{new}} s_{\mathrm{new}}$.

**Theorem 1.** *Suppose a PES $\mathit{\Gamma}$ for $P$ is $(d_1, d_2)$-admissible. Then, $\mathsf{CP1\text{-}Trans}(\mathit{\Gamma})$ for $\mathsf{CP1}[P]$ is $(\ell + m(d_1 - 1),\ md_2)$-admissible, where $m \times \ell$ is the size of policy.*

[9] Note that, since $\mathbf{s}'$ does not contain $s_0^{(i)}$, it is crucial that we use Corollary 1 where the linear combination relies only on $\tilde{\mathbf{s}}^{(i)} = (s_1^{(i)}, \ldots, s_{w_{1,i}}^{(i)})$.

*Proof.* We prove symbolic property of $\mathit{\Gamma}'$ from that of $\mathit{\Gamma}$ as follows.

**Selective Symbolic Property.** We define the following algorithms.

$\boxed{\mathsf{EncBS}'(\mathbf{A},\pi)}$: For each $i \in [m]$, run

$$\mathsf{EncBS}(\pi(i)) \to \Big(\mathbf{B}_1^{(i)},\ldots,\mathbf{B}_n^{(i)};\ \mathbf{s}_0^{(i)},\ldots,\mathbf{s}_{w_{1,i}}^{(i)};\ \hat{\mathbf{s}}_1^{(i)},\ldots,\hat{\mathbf{s}}_{w_{2,i}}^{(i)}\Big),$$

where $\mathbf{B}_j^{(i)} \in \mathbb{Z}_N^{d_1\times d_2}$, $\mathbf{s}_t^{(i)} \in \mathbb{Z}_N^{1\times d_2}$, $\hat{\mathbf{s}}_z^{(i)} \in \mathbb{Z}_N^{1\times d_1}$. For $j \in [2,n]$, we parse $\mathbf{B}_j^{(i)} =: \begin{pmatrix}\mathbf{e}_j^{(i)}\\ \tilde{\mathbf{B}}_j^{(i)}\end{pmatrix}$ where $\mathbf{e}_j^{(i)} \in \mathbb{Z}_N^{1\times d_2}$ and $\tilde{\mathbf{B}}_j^{(i)} \in \mathbb{Z}_N^{(d_1-1)\times d_2}$ (*i.e.*, decomposing into the first row and the rest). Let $d_1' = \ell + m(d_1-1)$ and $d_2' = md_2$. Any vector of length $d_2'$ can be naturally divided into $m$ blocks, each with length $d_2$. Any $d_1'$-length vectors consists of the first $\ell$ positions which are then followed by $m$ blocks of length $d_1-1$.[10] Let $\mathbf{B}_1' = \mathbf{1}_{1,1}^{d_1'\times d_2'}$, $\mathbf{s}_{\text{new}} = \mathbf{1}_1^{d_2'}$, $\mathbf{v}_\iota' = \mathbf{1}_\iota^{d_1'}$ for $\iota \in [2,\ell]$, and

$$\mathbf{B}_j' \;=\; \left(\begin{array}{cccc}
\mathbf{e}_j^{(1)}\mathbf{A}_{1,1} & \cdots & & \mathbf{e}_j^{(m)}\mathbf{A}_{m,1}\\
\vdots & & & \vdots\\
\mathbf{e}_j^{(1)}\mathbf{A}_{1,\ell} & \cdots & & \mathbf{e}_j^{(m)}\mathbf{A}_{m,\ell}\\
\hline
\tilde{\mathbf{B}}_j^{(1)} & & & \\
& \tilde{\mathbf{B}}_j^{(2)} & & \\
& & \ddots & \\
& & & \tilde{\mathbf{B}}_j^{(m)}
\end{array}\right) \qquad \in \mathbb{Z}_N^{d_1'\times d_2'}, \tag{13}$$

$$\mathbf{s}_t'^{(i)} = (0,\ldots,0,\ \overset{\substack{\text{block } i\\ \downarrow}}{\mathbf{s}_t^{(i)}}\ ,0,\ldots,0) \qquad \in \mathbb{Z}_N^{1\times d_2'},$$

$$\hat{\mathbf{s}}_z'^{(i)} = \big(\hat{\mathbf{s}}_z^{(i)}[1]\mathbf{A}_{i:},\ \ 0,\ldots,0,\ \overset{\substack{\text{block } i\\ \downarrow}}{\hat{\mathbf{s}}_z^{(i)}[2,d_1]},\ 0,\ldots,0\big) \in \mathbb{Z}_N^{1\times d_1'}, \tag{14}$$

for $j \in [2,n]$, $i \in [m]$, $t \in [w_{1,i}]$, $z \in [w_{2,i}]$. Output

$$\Big(\big(\mathbf{B}_j'\big)_{j\in[n]};\ \mathbf{s}_{\text{new}}, \big(\mathbf{s}_1'^{(i)},\ldots,\mathbf{s}_{w_{1,i}}'^{(i)}\big)_{i\in[m]};\ \mathbf{v}_2',\ldots,\mathbf{v}_\ell', \big(\hat{\mathbf{s}}_1'^{(i)},\ldots,\hat{\mathbf{s}}_{w_{2,i}}'^{(i)}\big)_{i\in[m]}\Big).$$

$\boxed{\mathsf{EncR}'(x,(\mathbf{A},\pi))}$: First note that we have the condition $\bar{P}_\kappa(x,(\mathbf{A},\pi)) = 0$. Let $S = \{\, i \in [m] \mid P_\kappa(x,\pi(i)) = 1 \,\}$.

1. From $\bar{P}_\kappa(x,(\mathbf{A},\pi)) = 0$ and from Proposition 1, we can obtain a vector $\boldsymbol{\omega} = (\omega_1,\ldots,\omega_\ell) \in \mathbb{Z}_N^{1\times\ell}$ such that $\omega_1 = 1$ and $\mathbf{A}_{i:}\boldsymbol{\omega}^\top = 0$ for all $i \in S$.
2. For each $i \notin S$, we can run $\mathsf{EncR}(x,\pi(i)) \to \Big(\mathbf{r}_1^{(i)},\ldots,\mathbf{r}_{m_1}^{(i)};\ \mathbf{a},\hat{\mathbf{r}}_1^{(i)},\ldots,\hat{\mathbf{r}}_{m_2}^{(i)}\Big)$, where $\mathbf{r}_v^{(i)} \in \mathbb{Z}_N^{1\times d_1}$, $\hat{\mathbf{r}}_u^{(i)} \in \mathbb{Z}_N^{1\times d_2}$, and $\mathbf{a} = \mathbf{1}_1^{d_2} \in \mathbb{Z}_N^{1\times d_2}$.

[10] That is, the $i$-th block of a vector $\mathbf{h} \in \mathbb{Z}_N^{1\times d_1'}$ is $\mathbf{h}[\ell+(d_1-1)(i-1)+1, \ell+(d_1-1)i]$.

3. For $i \in [m]$, let $g_i = \mathbf{A}_{i:}\boldsymbol{\omega}^\top / \mathbf{r}_v^{(i)}[1]$. Note that $\mathbf{r}_v^{(i)}[1] \neq 0$ due to admissibility.
4. Let $\mathbf{a}_{\text{new}} = \mathbf{1}_1^{d_2'}$, and for $v \in [m_1]$, $u \in [m_2]$ let

$$\mathbf{r}_v' = -\Big(\boldsymbol{\omega},\ g_1\mathbf{r}_v^{(1)}[2,d_1], \ldots, g_m\mathbf{r}_v^{(m)}[2,d_1]\Big) \in \mathbb{Z}_N^{1\times d_1'}, \tag{15}$$

$$\hat{\mathbf{r}}_u' = -(g_1\hat{\mathbf{r}}_u^{(1)}, \ldots, g_m\hat{\mathbf{r}}_u^{(m)}) \qquad \in \mathbb{Z}_N^{1\times d_2'}. \tag{16}$$

5. Output $(\mathbf{r}_1', \ldots, \mathbf{r}_{m_1}';\ \mathbf{a}_{\text{new}}, \hat{\mathbf{r}}_1', \ldots, \hat{\mathbf{r}}_{m_2}')$.

**Verifying Properties (**sketch**).** Properties (P1),(P3)-(P6) are straightforward. Due to limited space, we provide a sketch in verifying (P2)—zero evaluation of substituted polynomials—here, and defer the full details to the full version.

In $\mathsf{ct\text{-}enc}$ $\mathbf{c}'$, the $p$-th polynomial in $\mathbf{c}'^{(i)}$ is $c_p'^{(i)} =$

$$\sum_{z\in[w_{2,i}]} \eta_{p,z}^{(i)} \hat{s}_z'^{(i)} + \eta_{p,0,1}^{(i)}(\mathbf{A}_{i,1}b_1s_{\text{new}} + \sum_{\iota=2}^{\ell}\mathbf{A}_{i,\iota}v_\iota) + \sum_{\substack{t\in[w_{1,i}]\\ j\in[2,n]}} \eta_{p,t,j}^{(i)} b_j s_t'^{(i)}. \tag{17}$$

Substituting $\hat{s}_z'^{(i)} : (\hat{\mathbf{s}}_z'^{(i)})^\top$, $b_1s_{\text{new}} : \mathbf{B}_1'(\mathbf{s}_{\text{new}})^\top$, $v_\iota : (\mathbf{v}_\iota')^\top$, $b_js_t'^{(i)} : \mathbf{B}_j'(\mathbf{s}_t'^{(i)})^\top$, into $c_p'^{(i)}$ will result in a column vector of length $d_1' = \ell + m(d_1 - 1)$. We denote it as $\mathbf{w}^\top$. We claim that $\mathbf{w}^\top = 0$. We use the symbolic property of the base PES, $\mathit{\Gamma}$, which ensures that the substitution of $c_p^{(i)}$ via $\mathsf{EncBS}(\pi(i))$, denoted $\mathbf{u}^\top$, evaluates to 0. In fact, via elementary linear algebra, one can verify that for $j \in [\ell]$, $\mathbf{w}[j]$ is $\mathbf{u}[1]$ scaled by $\mathbf{A}_{i,j}$, and that the $i$-th block of $\mathbf{w}$ is exactly $\mathbf{u}[2, d_1]$, while the rest of $\mathbf{w}$ is already 0 by construction. Hence the claim holds.

In $\mathsf{key\text{-}enc}$ $\mathbf{k}$, the substitution for $k_1$ is straightforward. For $p \in [2, m_3]$, we have $k_p = \sum_{u\in[m_2]} \phi_{p,u}\hat{r}_u + \sum_{v\in[m_1],j\in[2,n]} \phi_{p,v,j}r_vb_j$. Substituting $\hat{r}_u : \hat{\mathbf{r}}_u'$, $r_vb_j : \mathbf{r}_v'\mathbf{B}_j'$ into $k_p$ will result in a row vector of length $d_2' = md_2$. We denote it as $\mathbf{w}$. We claim that $\mathbf{w} = 0$. Let $\mathbf{u}_i$ be the substitution result for $k_p$ via $\mathsf{EncR}(x, \pi(i))$. One can eventually verify that the $i$-th block of $\mathbf{w}$ is $g_i\mathbf{u}_i$, which evaluates to 0 since, if $i \in S$ we have $g_i = 0$, while if $i \notin S$ we have $\mathbf{u}_i = 0$ due to the symbolic property of the base PES. Hence the claim holds.

**Co-selective Symbolic Property.** Let $\mathsf{EncBR}'(x) = \mathsf{EncBR}(x)$.

$\boxed{\mathsf{EncS}'(x, (\mathbf{A}, \pi))}$: First note that we have the condition $\bar{P}_\kappa(x, (\mathbf{A}, \pi)) = 0$. Let $S = \{\, i \in [m] \mid P_\kappa(x, \pi(i)) = 1 \,\}$.

1. For each $i \notin S$, we have $P_\kappa(x, \pi(i)) = 0$. Thus, we can run $\mathsf{EncS}(x, \pi(i)) \to \Big(\mathbf{s}_0^{(i)}, \ldots, \mathbf{s}_{w_{1,i}}^{(i)};\ \hat{\mathbf{s}}_1^{(i)}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{(i)}\Big)$, where $\mathbf{s}_t^{(i)} \in \mathbb{Z}_N^{1\times d_2}$, and $\hat{\mathbf{s}}_z^{(i)} \in \mathbb{Z}_N^{1\times d_1}$.
2. From $\bar{P}_\kappa(x, (\mathbf{A}, \pi)) = 0$ and Proposition 1, we can obtain a vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_\ell)$ such that $\omega_1 = 1$ and $\mathbf{A}_{i:}\boldsymbol{\omega}^\top = 0$ for all $i \in S$. Let $q_i = \mathbf{A}_{i:}\boldsymbol{\omega}^\top$.
3. Let $\mathbf{s}_{\text{new}} = \mathbf{1}_1^{d_2}$, $\mathbf{s}_t'^{(i)} = q_i\mathbf{s}_t^{(i)}$, $\hat{\mathbf{s}}_z'^{(i)} = q_i\hat{\mathbf{s}}_z^{(i)}$, and $\mathbf{v}_\iota' = \omega_\iota\mathbf{1}_1^{d_1}$, for $i \in [m]$, $t \in [w_{1,i}]$, $\iota \in [2, \ell]$, $z \in [w_{2,i}]$.

4. Output $\left(\mathbf{s}_{\mathrm{new}}, \left(\mathbf{s}'^{(i)}_1, \ldots, \mathbf{s}'^{(i)}_{w_{1,i}}\right)_{i\in[m]}; \mathbf{v}'_2, \ldots, \mathbf{v}'_\ell, \left(\hat{\mathbf{s}}'^{(i)}_1, \ldots, \hat{\mathbf{s}}'^{(i)}_{w_{2,i}}\right)_{i\in[m]}\right)$.

**Verifying Properties.** First we can verify that $\mathbf{a}_{\mathrm{new}}\mathbf{s}_{\mathrm{new}}^\top = \mathbf{1}_1^{d_2}(\mathbf{1}_1^{d_2})^\top = 1 \neq 0$, as required. Next, since we define $\mathsf{EncBR}'(x) = \mathsf{EncBR}(x)$, the substitution for $\mathsf{key\text{-}enc}$ is trivially evaluated to 0, due to the co-selective symbolic property of $\mathit{\Gamma}$. It remains to consider the substitution for $\mathsf{ct\text{-}enc}$ $\mathbf{c}'$. For $i \in [m]$, $p \in [w_{3,i}]$, the polynomial $c_p^{(i)}$ is depicted in Eq. (17). We have that the middle sum term $\mathbf{A}_{i:}\mathbf{v}^\top$ is substituted and evaluated to $q_i(\mathbf{1}_1^{d_2})^\top$. Let $\mathbf{u}_i^\top \in \mathbb{Z}_N^{d_1\times 1}$ denote the substitution result for $c_p^{(i)}$ (as a part of $\mathbf{c}^{(i)}$) via $\mathsf{EncS}(x, \pi(i))$ (and $\mathsf{EncBR}(x)$). By our constructions of $\mathbf{s}'^{(i)}_t$ and $\hat{\mathbf{s}}'^{(i)}_z$, it is straightforward to see that the substitution for $c'^{(i)}_p$ (as a part of $\mathbf{c}'^{(i)}$) via $\mathsf{EncS}'(x, (\mathbf{A}, \pi))$ (and $\mathsf{EncBR}'(x)$) is indeed $q_i\mathbf{u}_i^\top$. Note that $\mathbf{u}_i^\top$ contains $\mathbf{B}_1\mathbf{s}_0^\top = \mathbf{1}_1^{d_2}$: this corresponds to the substitution of $\mathbf{A}_{i:}\mathbf{v}^\top$. Finally, we can see that $q_i\mathbf{u}_i^\top = 0$ since if $i \in S$ then $q_i = 0$, while if $i \notin S$, we have $\mathbf{u}_i^\top = 0$ due to the co-selective property of $\mathit{\Gamma}$. □

**Intuition.** Due to an abstract manner of our scheme, it might be useful to relate the above *selective* proof to the idea described in §2. Intuitively, the upper part of $\mathbf{B}'_j$ of Eq. (13) acts as a "projection", generalizing $\mathbf{B}'_j$ of Eq. (5) in §2, but now we also embed the policy $\mathbf{A}$ in a novel way. Consider the multiplication $\mathbf{r}'_v\mathbf{B}'_j$. Here, only "non-problematic" blocks (the $i$-th block where $i \notin S$) are turned "on" by $\boldsymbol{\omega}$ from $\mathbf{r}'_v$. All "problematic" blocks ($i \in S$) are turned "off" by the "mask" vector $(\mathbf{A}_{1:}\boldsymbol{\omega}^\top, \ldots, \mathbf{A}_{m:}\boldsymbol{\omega}^\top)$. We also note that this "mask" vector encodes the non-acceptance condition as per Proposition 1. All in all, this gives us the relation: $\mathbf{r}'_v\mathbf{B}'_j = -\big(g_1\mathbf{r}_v^{(1)}\mathbf{B}_j^{(1)}, \ldots, g_m\mathbf{r}_v^{(m)}\mathbf{B}_j^{(m)}\big)$, where we recover the substitution vectors of the base PES, namely, $\mathbf{r}_v^{(i)}\mathbf{B}_j^{(i)}$, and thus can use the base symbolic property. We succeed in doing so despite having the "projection" part, which seems to hinder the independency among blocks in the first place.

## 6 Key-policy Augmentation

For a predicate family $P$, we define its key-policy-span-program-augmented predicate—denoted as $\mathsf{KP1}[P]$—as the dual of $\mathsf{CP1}[P']$ where $P'$ is the dual of $P$. Therefore, we can use the dual conversion [10,2]—applying two times–sandwiching $\mathsf{CP1\text{-}Trans}$, to obtain a PES conversion for $\mathsf{KP1}[P]$. However, this would incur additional elements for encodings (from dual conversions). Below, we provide a direct conversion without additional elements.

**Construction 3.** Let $\mathit{\Gamma}$ be a PES construction for a $P$ satisfying admissibility. We construct a PES $\mathit{\Gamma}'$ for $\mathsf{KP1}[P]$ as follows. Denote this $\mathit{\Gamma}'$ by $\mathsf{KP1\text{-}Trans}(\mathit{\Gamma})$.

- $\mathsf{Param}'(\mathsf{par}) = \mathsf{Param}(\mathsf{par}) = n$. Denote $\mathbf{b} = (b_1, \ldots, b_n)$.
- $\mathsf{EncCt}'(y, N) = \mathsf{EncCt}(y, N) = \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b})$.

- $\mathsf{EncKey}'((\mathbf{A},\pi),N)$. Parse $\mathbf{A}\in\mathbb{Z}_N^{m\times\ell}$. Let $\mathbf{v}:=(\alpha_{\text{new}},v_2,\ldots,v_\ell)$ be new lone variables. For all $i\in[m]$, do as follows.
  - Run $\mathsf{EncKey}(\pi(i),N)$ to obtain a vector $\mathbf{k}^{(i)}=\mathbf{k}^{(i)}(\mathbf{r}^{(i)},\hat{\mathbf{r}}^{(i)},\mathbf{b})$ of polynomials in variables $\mathbf{r}^{(i)}=(r_1^{(i)},\ldots,r_{m_{1,i}}^{(i)})$, $\hat{\mathbf{r}}^{(i)}=(\alpha^{(i)},\hat{r}_1^{(i)},\ldots,\hat{r}_{m_{2,i}}^{(i)})$, $\mathbf{b}$.
  - Define a modified vector by variable replacement as

  $$\mathbf{k}'^{(i)}:=\mathbf{k}^{(i)}|_{\alpha^{(i)}\mapsto\mathbf{A}_{i:}\mathbf{v}^\top}.$$

  In fact, this only modifies $k_1^{(i)}=\alpha^{(i)}+r_1^{(i)}b_1$ to $k_1'^{(i)}=\mathbf{A}_{i:}\mathbf{v}^\top+r_1^{(i)}b_1$.

  Finally, output $\mathbf{k}'=\mathbf{k}'(\mathbf{r}',\hat{\mathbf{r}}',\mathbf{b})$ as $\mathbf{k}':=\big(\mathbf{k}'^{(i)}\big)_{i\in[m]}$. It contains variables $\mathbf{r}':=(\mathbf{r}^{(i)})_{i\in[m]}$, $\hat{\mathbf{r}}':=(\alpha_{\text{new}},v_2,\ldots,v_\ell,(\hat{\mathbf{r}}^{(i)})_{i\in[m]})$, and $\mathbf{b}$.

**Pair/Correctness.** For proving correctness, we suppose $\bar{P}_\kappa((\mathbf{A},\pi),y)=1$. Let $S:=\{\,i\in[m]\mid P_\kappa(\pi(i),y)=1\,\}$. For $i\in S$, we can run $\mathsf{Pair}(\pi(i),y,N)\to(\mathbf{E},\overline{\mathbf{E}})$ and obtain a linear combination $\mathbf{sE}(\mathbf{k}'^{(i)})^\top+\mathbf{c}\overline{\mathbf{E}}(\mathbf{r}^{(i)})^\top=\alpha^{(i)}s_0=\mathbf{A}_{i:}\mathbf{v}^\top s_0$. Now since $\mathbf{1}_1^\ell\in\text{span}(\mathbf{A}|_y)$, we have linear combination coefficients $\{\,t_i\,\}_{i\in S}$ such that $\sum_{i\in S}t_i\mathbf{A}_{i:}=\mathbf{1}_1^\ell$. Therefore, the above terms can be linearly combined to $\sum_{i\in S}t_i(\mathbf{A}_{i:}\mathbf{v}^\top)s_0=\alpha_{\text{new}}s_0$, as required.

**Theorem 2.** *Suppose a PES $\varGamma$ for $P$ is $(d_1,d_2)$-admissible. Then, the the PES* $\mathsf{KP1\text{-}Trans}(\varGamma)$ *for* $\mathsf{KP1}[P]$ *satisfies* $(md_1,\ m'd_2)$-$\mathsf{Sym\text{-}Prop}^+$*, where $m\times\ell$ is the size of policy and $m'=\max\{m,\ell\}$.*

The proof is analogous to CP1-Trans, and is deferred to the full version. Note that, unlike CP1-Trans, KP1-Trans does not preserve admissibility, by construction.

## 7 Direct Sum and Augmentation over Predicate Set

In this section, we explore policy augmentations over a *set* of predicate families. We will also introduce the *direct sum* predicate as an intermediate notion, which is of an independent interest in its own right.

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

**Definition 6.** We define the *key-policy-span-program-augmented predicate over set* $\mathcal{P}$ as $\mathsf{KP}[\mathcal{P}]=\{\,\bar{P}_\kappa\,\}_\kappa$ where $\bar{P}_\kappa:\bar{\mathcal{X}}_\kappa\times\bar{\mathcal{Y}}_\kappa\to\{\,0,1\,\}$ by letting

- $\bar{\mathcal{X}}_\kappa=\{\,(\mathbf{A},\pi)\mid\mathbf{A}\in\mathbb{M}(\mathbb{Z}_N),\ \pi:[m]\to\mathbb{X}_\kappa\,\}$.
- $\bar{\mathcal{Y}}_\kappa=2^{\mathbb{Y}_\kappa}$.

– $\bar{P}_\kappa((\mathbf{A},\pi),Y)=1 \iff \mathbf{1}_1^\ell \in \mathrm{span}(\mathbf{A}|_Y)$, where[11]

$$\mathbf{A}|_Y \coloneqq \left\{ \mathbf{A}_{i:} \;\middle|\; \exists(\pi_1(i),y)\in Y \text{ s.t. } P^{(\pi_1(i))}(\pi_2(i),\,y)=1 \right\}.$$

where $\pi(i)=(\pi_1(i),\pi_2(i))\in\mathbb{X}_\kappa$, and $m\times\ell$ is the size of the matrix $\mathbf{A}$. ♢

*Remark 1.* When $\mathcal{P}$ has one element, say $\mathcal{P}=\{P\}$, we abuse the notation and write $\mathsf{KP}[P]\coloneqq\mathsf{KP}[\{P\}]$. Note that $\mathsf{KP}[P]$ is still more powerful than $\mathsf{KP1}[P]$, defined in §6, as it allows a ciphertext attribute to be a set.

**Unbounded/Dynamic/Static/OR/AND.** We consider (confined) variants of the predicate $\mathsf{KP}[\mathcal{P}]$ as follows. We will confine the domain of $(\mathbf{A},\pi_1)$, which specifies a policy over predicates. Their full domain, inferred from Definition 6, is $D\coloneqq\bigcup_{m\in\mathbb{N}}\mathbb{M}_m(\mathbb{Z}_N)\times F_{m,k}$, where $F_{m,k}$ denotes the set of all functions that map $[m]$ to $[k]$. For a class $C\subseteq D$, the predicate $\mathsf{KP}[\mathcal{P}]$ with the domain of $(\mathbf{A},\pi_1)$ being confined to $C$ is denoted by $\mathsf{KP}_C[\mathcal{P}]$ and is also called *dynamic span-program composition with class $C$*. It is called *unbounded* if $C=D$. It is called *static* if $|C|=1$. We denote $\mathsf{KP}_{\mathsf{OR}}[\mathcal{P}]$ as the shorthand for $\mathsf{KP}_C[\mathcal{P}]$ where $C=\bigcup_{m\in\mathbb{N}}\{\mathbf{A}_{\mathsf{OR},m}\}\times F_{m,k}$, and call it the *key-OR-policy-augmented* predicate over $\mathcal{P}$. (Recall that $\mathbf{A}_{\mathsf{OR},m}$ is the matrix for the OR policy, see §3.4.) Analogous notations go for the cases of $\mathsf{KP1}_{\mathsf{OR}}$, $\mathsf{KP}_{\mathsf{AND}}$, $\mathsf{CP}_{\mathsf{OR}}$, and so on.

**Definition 7.** We define the predicate called the *direct sum of* $\mathcal{P}$ as $\mathsf{DS}[\mathcal{P}]=\{\,\bar{P}_\kappa\,\}_\kappa$ where we let the predicate be $\bar{P}_\kappa:\mathbb{X}_\kappa\times\mathbb{Y}_\kappa\to\{\,0,1\,\}$ with

$$\bar{P}_\kappa\big((i,x),\,(j,y)\big)=1 \iff \big(i=j\big)\;\wedge\;\big(P^{(j)}_{\kappa_j}(x,y)=1\big).$$

For notational convenience, we also denote it as $P^{(1)}\odot\cdots\odot P^{(k)}=\mathsf{DS}[\mathcal{P}]$. ♢

We are now ready to state a lemma for constructing $\mathsf{KP}[\mathcal{P}]$. The implication is quite straightforward from definitions. We defer the proof to the full version.

**Lemma 2.** $\mathsf{KP}[\mathcal{P}]$ *can be embedded into* $\mathsf{KP1}[\mathsf{CP1}_{\mathsf{OR}}[\mathsf{DS}[\mathcal{P}]]]$.

**Constructing PES for $\mathsf{KP}[\mathcal{P}]$.** Now, since $\mathsf{DS}[\mathcal{P}]$ is a *single* predicate family (rather than a *set* of them), we can apply the CP1-Trans and KP1-Trans to a PES for $\mathsf{DS}[\mathcal{P}]$ to obtain a PES for $\mathsf{KP}[\mathcal{P}]$. Note that we apply Layer-Trans for admissibility if necessary.

**Constructing PES for Direct Sum.** In the next two subsections, we provide two constructions of PESs for direct sum of a set $\mathcal{P}$ of predicate families. The first is a simpler one that simply "concatenates" all the base PESs for each predicate family in $\mathcal{P}$. The second is superior as the same parameter variables $\mathbf{b}$ can be "reused" for all predicate families in $\mathcal{P}$.

[11] In the bracket, we write $P^{(\pi_1(i))}$ instead of $P^{(\pi_1(i))}_{\kappa_{\pi_1(i)}}$ for simplicity.

### 7.1 Simple Direct Sum by Parameter Concatenation

**Construction 4.** Let $\mathit{\Gamma}^{(j)}$ be a PES for $P^{(j)}$. Also let $\mathbf{\Gamma} = (\mathit{\Gamma}^{(1)}, \ldots, \mathit{\Gamma}^{(k)})$. We construct a PES $\mathit{\Gamma}'$ for $\mathsf{DS}[\mathcal{P}]$, where $\mathcal{P} = \{P^{(1)}, \ldots, P^{(k)}\}$, as follows. For further use, we denote this $\mathit{\Gamma}'$ by $\mathsf{Concat\text{-}Trans}(\mathbf{\Gamma})$.

- $\mathsf{Param}'(\mathsf{par})$. For $j \in [k]$, run $\mathsf{Param}^{(j)}(\mathsf{par}_j)$ to obtain $n_j$. Denote $\mathbf{b}^{(j)} = (b_1^{(j)}, \ldots, b_{n_j}^{(j)})$. Output $n = n_1 + \ldots + n_k$. Denote $\mathbf{b}' = (\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(k)})$.
- $\mathsf{EncCt}'((j, y), N)$. Run $\mathsf{EncCt}^{(j)}(y, N) \to \mathbf{c} = \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}^{(j)})$ and output $\mathbf{c}$.
- $\mathsf{EncKey}'((i, x), N)$. Run $\mathsf{EncKey}^{(i)}(x, N) \to \mathbf{k} = \mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}^{(i)})$ and output $\mathbf{k}$.

**Pair/Correctness.** This is straightforward from the base schemes. More precisely, for proving correctness, we suppose $\bar{P}_\kappa\big((i, x), (j, y)\big) = 1$. That is, $i = j$ and $P^{(j)}_{\kappa_j}(x, y) = 1$. Hence, we can run $\mathsf{Pair}^{(j)}(x, y, N) \to (\mathbf{E}, \overline{\mathbf{E}})$ and obtain a linear combination $\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \alpha s_0$, as required.

To prove symbolic security of $\mathsf{Concat\text{-}Trans}(\mathbf{\Gamma})$, we use one more intermediate constraint for the underlying PESs, called $\mathsf{Sym\text{-}Prop}^{++}$, which, in turn, can be converted from PES with normal $\mathsf{Sym\text{-}Prop}$ via $\mathsf{Plus\text{-}Trans}$. We defer these proofs to the full version. Below, we let $\perp$ be a special symbol which is not in $\mathcal{Y}_\kappa$, $\mathcal{X}_\kappa$, and abuse notation by letting any predicate evaluate to 0 if at least one input is the symbol $\perp$.

**Definition 8.** A PES $\mathit{\Gamma}$ for predicate family $P$ satisfies $(d_1, d_2)$-$\mathsf{Sym\text{-}Prop}^{++}$ if it satisfies $(d_1, d_2)$-$\mathsf{Sym\text{-}Prop}^{+}$ with the following further requirement.

(P7). In the selective symbolic property definition, the zero evaluation property of $\mathsf{key\text{-}enc}$ (P2) also holds for $\mathsf{EncB}(\perp)$, $\mathsf{EncR}(x, \perp)$ for all $x \in \mathcal{X}_\kappa$. $\diamondsuit$

**Lemma 3.** *Suppose that, for all $j \in [k]$, the PES $\mathit{\Gamma}^{(j)}$ for predicate family $P^{(j)}$ satisfies $(d_1, d_2)$-$\mathsf{Sym\text{-}Prop}^{++}$. Then, the PES $\mathsf{Concat\text{-}Trans}(\mathbf{\Gamma})$ for predicate family $\mathsf{DS}[\mathcal{P}]$, where $\mathcal{P} = \{P^{(1)}, \ldots, P^{(k)}\}$, satisfies $(d_1, d_2)$-$\mathsf{Sym\text{-}Prop}^{+}$.*

### 7.2 Efficient Direct Sum with Parameter Reuse

**Construction 5.** Let $\mathit{\Gamma}^{(j)}$ be a PES for $P^{(j)}$. Also let $\mathbf{\Gamma} = (\mathit{\Gamma}^{(1)}, \ldots, \mathit{\Gamma}^{(k)})$. We construct a PES $\mathit{\Gamma}'$ for $\mathsf{DS}[\mathcal{P}]$, where $\mathcal{P} = \{P^{(1)}, \ldots, P^{(k)}\}$, as follows. We denote this scheme by $\mathsf{Reuse\text{-}Trans}(\mathbf{\Gamma})$. The intuition is to use two new parameters $g_j, h_j$ specific to $\mathit{\Gamma}^{(j)}$, where in the proof, their substituted matrices serve as the "switches" that turn on only the $j$-th scheme, and that is why we can reuse the same based parameters $\mathbf{b}$ (since the others are rendered zero by the switches).

- $\mathsf{Param}'(\mathsf{par})$. For $j \in [k]$, run $\mathsf{Param}^{(j)}(\mathsf{par}_j)$ to obtain $n_j$. Let $n = \max_{j \in [k]} n_j$. Output $n' = n + 2k$. Denote $\mathbf{b} = (b_1, \ldots, b_n, g_1, \ldots, g_k, h_1, \ldots, h_k)$. Also denote $\mathbf{b}_j = (b_1, \ldots, b_{n_j})$.

- $\mathsf{EncCt}'((j,y),N)$. Run $\mathsf{EncCt}^{(j)}(y,N) \to \mathbf{c} = \mathbf{c}(\mathbf{s},\hat{\mathbf{s}},\mathbf{b}_j)$. Let $s_{\rm new}$ be the new special non-lone variable. Output $\mathbf{c}' = \big(\,\mathbf{c},\quad g_j s_0 + h_j s_{\rm new}\,\big)$.
- $\mathsf{EncKey}'((i,x),N)$. Run $\mathsf{EncKey}^{(i)}(x,N) \to \mathbf{k} = \mathbf{k}(\mathbf{r},\hat{\mathbf{r}},\mathbf{b}_i)$. Let $r_{\rm new}$ be a new non-lone variable and $\alpha_{\rm new}$ be the new special lone variable. Let $\tilde{\mathbf{k}}$ be exactly $\mathbf{k}$ but with $\alpha$ being replaced by $r_{\rm new}g_i$. Output $\mathbf{k}' = \big(\,\tilde{\mathbf{k}},\quad \alpha_{\rm new} + r_{\rm new}h_i\,\big)$.

**Pair/Correctness.** Suppose $\bar{P}_\kappa\big((i,x),\,(j,y)\big) = 1$. Thus, $i = j$ and $P^{(j)}_{\kappa_j}(x,y) = 1$. Hence, we can run $\mathsf{Pair}^{(j)}(x,y,N) \to (\mathbf{E},\overline{\mathbf{E}})$ and obtain a linear combination $\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \alpha s_0 = (r_{\rm new}g_j)s_0$. Hence, we have the following, as required: $\big(\,\alpha_{\rm new} + r_{\rm new}h_j\,\big)s_{\rm new} - r_{\rm new}\big(\,g_j s_0 + h_j s_{\rm new}\,\big) + (r_{\rm new}g_j)s_0 = \alpha_{\rm new}s_{\rm new}$.

**Lemma 4.** *Suppose that PES $\Gamma^{(j)}$ for $P^{(j)}$ satisfies $(d_1,d_2)$-*$\mathsf{Sym\text{-}Prop}^+$*, for all $j \in [k]$. Then, the PES* $\mathsf{Reuse\text{-}Trans}(\mathbf{\Gamma})$ *for predicate family* $\mathsf{DS}[\mathcal{P}]$*, where $\mathcal{P} = \{P^{(1)},\dots,P^{(k)}\}$, satisfies $(d_1,d_2)$-*$\mathsf{Sym\text{-}Prop}^+$*.* (The proof is deferred to the full version.)

## 8 Predicative Automata

This section presents an augmentation via DFA over predicates. Due to direct sum transformations, it is again sufficient to consider a single predicate variant.

Let $P = \{\,P_\kappa\,\}_\kappa$ where $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{\,0,1\,\}$, be a predicate family. A *Predicative Automata* (PA) over $P_\kappa$ is a 4-tuple $(Q,\mathcal{T},q_0,F)$ where $Q$ is the set of states, $\mathcal{T} \subseteq Q \times Q \times \mathcal{X}_\kappa$ is the transition table, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of accept states. For simplicity and w.l.o.g., we can assume that there is only one accept state, and it has no outgoing transition. An input to such an automata is a sequence $Y = (y_1,\dots,y_\ell) \in (\mathcal{Y}_\kappa)^*$, where $\ell$ is unbounded. A predicative automata $M = (Q = \{q_0,\dots,q_{\sigma-1}\},\mathcal{T},q_0,q_{\sigma-1})$ accepts $Y$ if there exists a sequence of states $(q^{(1)},\dots,q^{(\ell)}) \in Q^\ell$ such that for all $i \in [1,\ell]$, it holds that there exists $(q^{(i-1)},q^{(i)},x^{(i)}) \in \mathcal{T}$ such that $P_\kappa(x^{(i)},y_i) = 1$, and that $q^{(0)} = q_0$ and $q^{(\ell)} = q_{\sigma-1}$. Following the predicate for deterministic finite automata (DFA) [35,5,2], we will assume *determinism* of such a predicative automata. (So we may call it predicative DFA.) In our context, this is the restriction that for any different transitions with the same outgoing state, namely $(q,q',x')$ and $(q,q'',x'')$ with $q' \neq q''$, we require that for all $y \in \mathcal{Y}_\kappa$, it must be that $P_\kappa(x',y) \neq P_\kappa(x'',y)$. We can observe that if $P$ is the equality predicate (IBE), then the resulting predicative DFA over $P$ is exactly the definition of DFA.

**Example.** We provide an example of languages. Suppose we have a list of words which are considered BAD. There exists a simple predicative DFA, depicted in Fig. 5, that accepts exactly any sentences that start with a BAD word and contain an even number of the total BAD words. This seems not possible with span programs, since a sentence can be arbitrarily long.

**Definition 9.** Let $P = \{\,P_\kappa\,\}_\kappa$ where $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{\,0,1\,\}$, be a predicate family, indexed by $\kappa = (N,\mathsf{par})$. We define the *Key-policy-Automata-augmented predicate* over $P$ as $\mathsf{KA1}[P] = \big\{\,\bar{P}_\kappa\,\big\}_\kappa$ where $\bar{P}_\kappa : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{\,0,1\,\}$ by letting

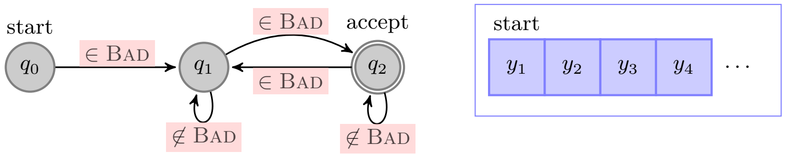


Fig. 5: Predicative DFA for language of sentences that start with a bad word and have an even number of the total bad words. Based predicates for testing membership/non-membership can use IBBE, IBR, defined in §9.2, respectively.

- $\bar{\mathcal{X}}_\kappa = \{\, M \mid M$ is a predicative automata over $P_\kappa\ \}$.
- $\bar{\mathcal{Y}}_\kappa = (\mathcal{Y}_\kappa)^*$.
- $\bar{P}_\kappa(M, Y) = 1 \iff M$ accepts $Y$. ◊

**Intuition.** The intuition for constructing PESs for DFA over predicates is similar to that of span program over predicates in that we follow the blueprint of generalizing PESs for X over IBE to X over any predicates, where X is either DFA or span program. Note that this blueprint was explained in §2 for the case of span programs. Here, for the DFA case, the starting PES is the ABE for regular languages (which can be considered as DFA over IBE) of [5], of which a symbolic proof was given in §B.5 of [2]. In our construction below, one may notice that the structure of PES contains "two copies" of the underlying PES. This feature is inherited from the PES for ABE for regular languages of [5], which already utilizes two copies of IBE encodings.

We note some differences from the case of span programs. For the constructions, while our conversions for span programs use the second approach in §2 (based on admissible PES), we will base our conversion for DFA instead on the first approach (using the layering technique). This is done for simplicity. For the proofs, we note that span programs and DFAs have completely different combinatorial properties and thus different kinds of substituted matrices. See more discussions below.

**Construction 6.** Let $\Gamma$ be a PES construction for $P$. We construct a PES $\Gamma'$ for $\mathsf{KA1}[P]$ as follows. For further use, we denote this $\Gamma'$ by $\mathsf{KA1\text{-}Trans}(\Gamma)$.

- $\mathsf{Param}'(\mathsf{par})$. If $\mathsf{Param}(\mathsf{par})$ returns $n$, then output $2n + 5$. Denote $\mathbf{b}_1 = (b_{1,1}, \ldots, b_{1,n})$, $\mathbf{b}_2 = (b_{2,1}, \ldots, b_{2,n})$, and $\mathbf{b}' = (\mathbf{b}_1, \mathbf{b}_2, h_0, g_1, h_1, g_2, h_2)$.
- $\mathsf{EncCt}'(Y, N)$. Parse $Y = (y_1, \ldots, y_\ell)$. For $i \in [\ell]$, run $\mathsf{EncCt}(y_i, N)$ to obtain a vector $\mathbf{c}^{(i)}$ of polynomials. We will use two copies of it, with two different sets of variables, written as:

$$\mathbf{c}^{(1,i)} := \mathbf{c}^{(i)}(\mathbf{s}^{(1,i)}, \hat{\mathbf{s}}^{(1,i)}, \mathbf{b}_1), \qquad \mathbf{c}^{(2,i)} := \mathbf{c}^{(i)}(\mathbf{s}^{(2,i)}, \hat{\mathbf{s}}^{(2,i)}, \mathbf{b}_2),$$

and relate these two sets of variables via:

$$s_0'^{(i)} := \begin{cases} s_0^{(1,i+1)} & \text{if } i = 0 \\ s_0^{(1,i+1)} = s_0^{(2,i)} & \text{if } i = 1, \ldots, \ell - 1 \\ s_0^{(2,i)} & \text{if } i = \ell \end{cases} . \tag{18}$$

We then define $c'_0 := h_0 s^{(0)}_{\mathrm{new}}$ and, for $i \in [\ell]$,

$$c'_i := h_1 s^{(i-1)}_{\mathrm{new}} + g_1 s'^{(i-1)}_0 + h_2 s^{(i)}_{\mathrm{new}} + g_2 s'^{(i)}_0,$$

where $s^{(0)}_{\mathrm{new}}, \dots, s^{(\ell)}_{\mathrm{new}}$ are new non-lone variables with $s^{(\ell)}_{\mathrm{new}}$ being special. Finally, it outputs $\mathbf{c}' := \big( c'_0, c'_1, \dots, c'_\ell, \, ( \, \mathbf{c}^{(1,i)}, \mathbf{c}^{(2,i)} \, )_{i \in [\ell]} \, \big)$.

- $\mathsf{EncKey}'(M, N)$. Parse $M = (Q, \mathcal{T}, q_0, q_{\sigma-1})$ and parse $\mathcal{T} = \big\{ \, (q_{\upsilon_t}, q_{\omega_t}, x_t) \, \big\}_{t \in [m]}$ where each $\upsilon_t, \omega_t \in [0, \sigma - 1]$. [12] Let $u_0, u_1, \dots, u_{\sigma-1}$ be new lone variables with $u_{\sigma-1}$ being special. For all $t \in [m]$, run $\mathsf{EncKey}(x_t, N)$ to obtain a vector $\mathbf{k}^{(t)}$ of polynomials. We use two copies of it, with two different sets of variables. We then modify them via variable replacement as follows.

$$\begin{aligned} \mathbf{k}^{(1,t)} &:= \mathbf{k}^{(t)}(\mathbf{r}^{(1,t)}, \hat{\mathbf{r}}^{(1,t)}, \mathbf{b}_1), & \mathbf{k}^{(2,t)} &:= \mathbf{k}^{(t)}(\mathbf{r}^{(2,t)}, \hat{\mathbf{r}}^{(2,t)}, \mathbf{b}_2), \\ \mathbf{k}'^{(1,t)} &:= \mathbf{k}^{(1,t)}|_{\alpha^{(1,t)} \mapsto r^{(t)}_{\mathrm{new}} g_1}, & \mathbf{k}'^{(2,t)} &:= \mathbf{k}^{(2,t)}|_{\alpha^{(2,t)} \mapsto r^{(t)}_{\mathrm{new}} g_2}, \end{aligned}$$

where $r^{(t)}_{\mathrm{new}}$ is a new non-lone variable (the same one for both). We then define

$$\tilde{k}_0 := -u_0 + r^{(0)}_{\mathrm{new}} h_0, \quad \tilde{k}_{1,t} := u_{\upsilon_t} + r^{(t)}_{\mathrm{new}} h_1, \quad \tilde{k}_{2,t} := -u_{\omega_t} + r^{(t)}_{\mathrm{new}} h_2.$$

for $t \in [m]$. Finally, it outputs $\mathbf{k}' := \Big( \tilde{k}_0, \big( \, \tilde{k}_{1,t}, \tilde{k}_{2,t}, \mathbf{k}'^{(1,t)}, \mathbf{k}'^{(2,t)}, \, \big)_{t \in [m]} \Big)$.

**Pair/Correctness.** Suppose $\bar{P}_\kappa(M, Y) = 1$. That is, there exists a sequence $(q^{(1)}, \dots, q^{(\ell)}) \in Q^\ell$ such that for all $i \in [1, \ell]$, it holds that $P_\kappa(x^{(i)}, y_i) = 1$ and $(q^{(i-1)}, q^{(i)}, x^{(i)}) \in \mathcal{T}$, and that $q^{(0)} = q_0$, while $q^{(\ell)} = q_{\sigma-1}$. For $i \in [\ell]$, we proceed as follows. Denote $t_i \in [m]$ as the transition index that corresponds to the $i$-th move; that is, let $(q_{\upsilon_{t_i}}, q_{\omega_{t_i}}, x_{t_i}) = (q^{(i-1)}, q^{(i)}, x^{(i)})$. From this, we have $q_{\upsilon_{t_i}} = q_{\omega_{t_{i-1}}}$ for all $i \in [\ell]$. Now since $P_\kappa(x_{t_i}, y_i) = 1$, we can run $\mathsf{Pair}(x_{t_i}, y_i, N)$ to obtain linear combinations that are equal to

$$\begin{aligned} D_{1,i} &:= \alpha^{(1,t_i)} s^{(1,i)}_0 = \big( r^{(t_i)}_{\mathrm{new}} g_1 \big) s'^{(i-1)}_0, \\ D_{2,i} &:= \alpha^{(2,t_i)} s^{(2,i)}_0 = \big( r^{(t_i)}_{\mathrm{new}} g_2 \big) s'^{(i)}_0. \end{aligned}$$

We have $Q_i := D_{1,i} + D_{2,i} + s^{(i-1)}_{\mathrm{new}} \tilde{k}_{1,t_i} + s^{(i)}_{\mathrm{new}} \tilde{k}_{2,t_i} - c'_i r^{(t_i)}_{\mathrm{new}} = s^{(i-1)}_{\mathrm{new}} u_{\omega_{t_{i-1}}} - s^{(i)}_{\mathrm{new}} u_{\omega_{t_i}}$. Let $Q_0 := s^{(0)}_{\mathrm{new}} \tilde{k}_0 - r^{(0)}_{\mathrm{new}} c'_0 = -s^{(0)}_{\mathrm{new}} u_0$. Combining them, we obtain $-\sum_{i=0}^{\ell} Q_i = s^{(\ell)}_{\mathrm{new}} u_{\sigma-1}$, as required.

**Theorem 3.** *Suppose a PES $\Gamma$ for $P$ satisfies $(d_1, d_2)$-$\mathsf{Sym\text{-}Prop}^{++}$. Then, the the PES $\mathsf{KA1\text{-}Trans}(\Gamma)$ for $\mathsf{KA1}[P]$ satisfies $(\psi_1 d_1, \, \psi_2 d_2)$-$\mathsf{Sym\text{-}Prop}^{+}$, where $\psi_1 = \max\{\ell + 1, m\}$, $\psi_2 = \max\{\ell + 1, 2m\}$, where $\ell$ is the size of ciphertext attribute $Y$, and $m$ is the size of transition table $\mathcal{T}$ for predicative automata $M$.*

We defer the proof to the full version. At the core, we point out combinatorial vectors that encode the non-acceptance condition of predicative DFA and use them as the "mask" vectors in the proof. Since the combinatorial properties here is richer than the KP1 case, the proof is somewhat more complex.

[12] $\upsilon_t, \omega_t$ indicate the "from" and the "to" state of the $t$-th transition in $\mathcal{T}$, respectively.

# 9 Applications

We provide applications from our framework. Due to limited space, we offer more discussions in the full version, where we also motivate for real-world applications.

## 9.1 ABE for New Predicates

**Predicative Branching Program.** This is similar to and might be less powerful than predicative DFA but may serve an independent interest, since its definition and construction are simpler. A *Predicative Branching Program* (PBP) over a predicate $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$ is a 4-tuple $(\Gamma, q_1, q_\sigma, L)$ where $\Gamma = (V, E)$ is a directed acyclic graph (DAG) with a set of nodes $V = \{q_1, \ldots, q_\sigma\}$ and a set of directed edges $E \subseteq V^2$, $q_1$ is a distinguished terminal node (a node with no outgoing edge) called the accept node, $q_\sigma$ is the unique start node (the node with no incoming edge), and $L : E \to \mathcal{X}_\kappa$ is an edge labelling function. An input to a PBP $M = (\Gamma, q_1, q_\sigma, L)$ is $y \in \mathcal{Y}_\kappa$. Let $\Gamma_y$ be an induced subgraph of $\Gamma$ that contains exactly all the edges $e$ such that $P_\kappa(L(e), y) = 1$. Such a PBP $M$ accepts $y$ if $\Gamma_y$ contains a directed path from the start node, $q_\sigma$, to the accept node, $q_1$. Following the deterministic characteristic of boolean branching programs, we will assume *determinism* of PBP: for any node $v$, for any two outgoing edges $e_1, e_2$ from the same node $v$, we require that $P_\kappa(L(e_1), y) \neq P_\kappa(L(e_2), y)$ for any $y \in \mathcal{Y}_\kappa$. We denote the key-policy-augmented predicate using PBP over $\mathcal{P}$ as $\mathsf{KB1}[\mathcal{P}]$. We show that it can be embedded into $\mathsf{KP1}[\mathcal{P}]$ by using almost the same proof as in the case for the implication ABE for span programs to ABE for BP in [6]. We provide this in the full version.

**Nested-policy/Mixed-policy ABE.** We can define new type of ABE that nests policies. Nested-policy ABE is ABE for predicate $\mathsf{CP}[\mathsf{KP}[\mathcal{P}]]$ or $\mathsf{KP}[\mathsf{CP}[\mathcal{P}]]$, or any arbitrarily hierarchically nested ones. In these schemes, however, the *structure of nesting* is fixed. We define what we call *Mixed-policy ABE* to free up this restriction altogether. It is defined in a recursive manner to make sure that at level $\ell$, it includes all the possible nesting structures that have at most $\ell$ layers. To construct a transformation for this, we observe that a trivial scheme using *parameter concatenation* would be inefficient as when going from level $\ell - 1$ to $\ell$, the number of parameters will become at least $d$ times of level $\ell - 1$, where $d$ is the number of transformations plus one (*e.g.,* if we want only $\mathsf{KP}[\cdot]$ and $\mathsf{CP}[\cdot]$, then $d = 3$). Hence, the overall size at level $\ell$ would be $O(d^\ell)$. Fortunately, thanks to our construction for direct sum with *parameter reuse*, $\mathsf{Reuse\text{-}Trans}$, the parameter size (which will correspond to the public key size for ABE) can be kept small. For $\ell$-level scheme, the parameter size is $O(n + k + d\ell)$, where $n$ is the maximum parameter size among $k$ based predicates in $\mathcal{P}$. We explore this in more details in the full version.

## 9.2 Revisiting Known Predicates

**Known Predicates and Modular Constructions.** We describe some known predicates and how they are related to more basic predicates via the policy augmented predicate notions (*e.g.,* $\mathsf{KP1}[\cdot]$, $\mathsf{KP}[\cdot]$). These relations directly suggest what transformations (*e.g.,* $\mathsf{KP1\text{-}Trans}$) can be used so as to achieve PES for more expressive predicates from only PESs for basic predicates, namely, IBE and its negation (NIBE), in a modular way. We note that the ciphertext-policy variants can be considered analogously, and can be obtained simply by applying the dual conversion [5,2]. Let $\mathcal{U} = \mathbb{Z}_N$ be the attribute universe.

We consider the following predicates.

- $P^{\mathsf{IBE}} : \mathcal{U} \times \mathcal{U} \to \{0,1\}$ is defined as $P^{\mathsf{IBE}}(x,y) = 1 \Leftrightarrow x = y$.
- $P^{\mathsf{NIBE}} : \mathcal{U} \times \mathcal{U} \to \{0,1\}$ is defined as $P^{\mathsf{NIBE}}(x,y) = 1 \Leftrightarrow x \neq y$.
- $P^{\mathsf{IBBE}} : \mathcal{U} \times 2^{\mathcal{U}} \to \{0,1\}$ is defined as $P^{\mathsf{IBBE}}(x,Y) = 1 \Leftrightarrow x \in Y$.[13]
  - It is clear that $P^{\mathsf{IBBE}}$ can be embedded into $\mathsf{CP1}_{\mathsf{OR}}[P^{\mathsf{IBE}}]$.
- $P^{\mathsf{IBR}} : \mathcal{U} \times 2^{\mathcal{U}} \to \{0,1\}$ is defined as $P^{\mathsf{IBR}}(x,Y) = 1 \Leftrightarrow x \notin Y$.
  - It is clear that $P^{\mathsf{IBR}}$ can be embedded into $\mathsf{CP1}_{\mathsf{AND}}[P^{\mathsf{NIBE}}]$.
- $P^{\mathsf{TIBBE}} : (\{1,2\} \times \mathcal{U}) \times 2^{\mathcal{U}} \to \{0,1\}$ is defined as $P^{\mathsf{TIBBE}}((i,x),Y) = 1 \Leftrightarrow (i = 1 \wedge x \in Y) \vee (i = 2 \wedge x \notin Y)$.[14]
  - It is clear that $P^{\mathsf{TIBBE}}$ can be embedded into $\mathsf{CP1}_{\mathsf{OR}}[P^{\mathsf{IBBE}} \odot P^{\mathsf{IBR}}]$.
- The predicate for completely-unbounded KP-ABE for monotone span program $P^{\mathsf{KP\text{-}MSP}}$ (as defined in [5] and recapped in §3.4) is the same as $\mathsf{KP1}[P^{\mathsf{IBBE}}]$, or equivalently, $\mathsf{KP}[P^{\mathsf{IBE}}]$.
- The predicate for completely-unbounded KP-ABE for non-monotone span program $P^{\mathsf{KP\text{-}NSP}}$ corresponds to exactly the definition of $\mathsf{KP1}[P^{\mathsf{TIBBE}}]$.

For self-containment, we provide PES constructions for $P^{\mathsf{IBE}}$ and $P^{\mathsf{NIBE}}$ in the full version.

**On ABE for Non-monotone Span Programs.** To the best of our knowledge, fully secure completely-unbounded large-universe KP-ABE for non-monotone span program (NSP) had not been achieved before this work. We achieve a scheme in prime-order groups, in a modular and clean manner from simple PESs for $P^{\mathsf{IBE}}$ and $P^{\mathsf{NIBE}}$. An explicit description of our PES for it is given in the full version. We have to rely on the q-ratio assumption, inherited from the framework of [2][15]; nevertheless, all the current *completely unbounded* ABE for even *monotone* span programs still also need q-type assumptions [31,5,2], even *selectively* secure one [31]. We provide a comparison to known KP-ABE schemes for NSP in prime-order groups in Table 1. We further discuss why large-universe ABE for NSP is generally a more difficult task to achieve than ABE for MSP in the full version.

For the CP-ABE case, a fully secure completely-unbounded scheme for NSP was recently and independently reported in [39]. Their scheme is constructed in composite-order groups. Our instantiated CP-ABE for NSP is in prime-order

[13] IBBE is for ID-based broadcast encryption [19]; IBR is for ID-based revocation [9].

[14] This is a unified notion for IBBE and IBR, and is called two-mode IBBE in [38].

[15] In defense, we also provide a positive remark towards the q-ratio assumption in the full version.

Table 1: Summary for KP-ABE for non-monotone span programs with large universe.

| Schemes | | \|PK\| | \|SK\| | \|CT\| | Unbounded | | | Security | Assumption |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | \|policy\|/ | multi-use/ | \|attrib. set\| | | |
| OSW07 [28] | I | $O(T)$ | $O(m)$ | $O(T)$ | ✓ | ✓ | | selective | DBDH |
| | II | $O(T)$ | $O(m\log(T))$ | $O(t)$ | ✓ | ✓ | | selective | DBDH |
| OT10 [29] | | $O(TR)$ | $O(m)$ | $O(tR)$ | ✓ | | | full | DLIN |
| OT12 [30] | | $O(1)$ | $O(m)$ | $O(tR)$ | ✓ | | ✓ | full | DLIN |
| ALP11 [9] | | $O(T)$ | $O(Tm)$ | $O(1)$ | ✓ | ✓ | | selective | $T$-DBDHE$^\dagger$ |
| YAHK14 [38] | I | $O(T)$ | $O(Tm)$ | $O(1)$ | ✓ | ✓ | | selective | $T$-DBDHE$^\dagger$ |
| | II | $O(T)$ | $O(m)$ | $O(T)$ | ✓ | ✓ | | selective | DBDH |
| | III | $O(T)$ | $O(m\log(T))$ | $O(t)$ | ✓ | ✓ | | selective | DBDH |
| | IV | $O(1)$ | $O(m)$ | $O(t)$ | ✓ | ✓ | ✓ | selective | $t$-A$^\dagger$ |
| Our KP-NSP | I | $O(1)$ | $O(m)$ | $O(t)$ | ✓ | ✓ | ✓ | full | qratio$^\dagger$ |
| | II | $O(T^2)$ | $O(T^3m)$ | $O(1)$ | ✓ | ✓ | | full | qratio$^\dagger$ |
| | III | $O(M^2+ML)$ | $O(1)$ | $O(t(M^3+M^2L))$ | | ✓ | ✓ | full | qratio$^\dagger$ |

Note: $t = |\text{attribute set}|$, $m \times \ell$ is the span program size, $R$ is the attribute multi-use bound, $T, M, L$ are the maximum bound for $t, m, \ell$, respectively (if required). Assumptions with $^\dagger$ are q-type assumptions.

groups, and unlike [39] of which proof is complex and specific, ours can be obtained in a modular manner. We defer a comparison table for CP-ABE for NSP to the full version.

**On Constant-size Schemes.** One huge further advantage in using the symbolic PES framework of [2] is that any symbolically secure PES can be transformed to constant-size schemes (in ciphertext or key sizes) by bounding corresponding terms and trading-off with the parameter size ($n$ from $\mathsf{Param}$). In particular, any of our transformed PESs in this paper, *e.g.,* $\mathsf{KP}[\mathcal{P}]$, can be made constant-size. We include such ABE for NSP in Table 1. More discussions on their detail complexities are in the full version.

**Revisiting the Okamoto-Takashima Definition.** The Okamoto-Takashima type ABE [29,30] for non-monotone span program was defined differently. We recast it here in our terminology, and explain how to achieve a PES for it in a modular manner in the full version.

**Acknowledgement.** This work was partially supported by JST CREST Grant No. JPMJCR1688.

## References

1. S. Agrawal, M. Chase. A Study of Pair Encodings: Predicate Encryption in Prime Order Groups. In *TCC 2016-A*, *LNCS*, pp. 259–288, 2016.
2. S. Agrawal, M. Chase. Simplifying Design and Analysis of Complex Predicate Encryption Schemes. In *Eurocrypt 2017*, *LNCS*, pp. 627–656, 2017.
3. M. Ambrona, G. Barthe, B. Schmidt. Generic Transformations of Predicate Encodings: Constructions and Applications. In *Crypto (1) 2017*, *LNCS*, pp. 36–66, 2017.
4. N. Attrapadung, H. Imai. Dual-Policy Attribute Based Encryption. In *ACNS 2009*, *LNCS*, pp. 168–185, 2009.

5. N. Attrapadung. Dual System Encryption via Doubly Selective Security: Framework, Fully-secure Functional Encryption for Regular Languages, and More. In *Eurocrypt 2014*, *LNCS*, pp. 557–577, 2014.
6. N. Attrapadung. Dual System Encryption Framework in Prime-Order Groups via Computational Pair Encodings. In *Asiacrypt 2016*, *LNCS*, pp. 591–623, 2016.
7. N. Attrapadung, G. Hanaoka, S. Yamada. Conversions among Several Classes of Predicate Encryption and Applications to ABE with Various Compactness Tradeoffs. In *Asiacrypt 2015*, *LNCS*, pp. 575–601, 2015.
8. N. Attrapadung, G. Hanaoka, K. Ogawa, G. Ohtake, H. Watanabe, S. Yamada. Attribute-Based Encryption for Range Attributes. In *SCN'16*, *LNCS*, pp. 42–61, 2016.
9. N. Attrapadung, B. Libert, E. Panafieu. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In *PKC 2011*, *LNCS*, pp. 90–108.
10. N. Attrapadung, S. Yamada. Duality in ABE: Converting Attribute Based Encryption for Dual Predicate and Dual Policy via Computational Encodings. In *CT-RSA 2015*, *LNCS*, pp. 87–105, 2015.
11. A. Beimel. Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
12. J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. In *IEEE S&P 2007*, pp. 321–334, 2007.
13. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Journal of Cryptology*, 24 (4), pp. 659–693, 2011. Extended abstract in *Eurocrypt 2004*, *LNCS*, pp. 223–238, 2004.
14. D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt 2008*, *LNCS*, pp. 455–470, 2008.
15. D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC 2011*, *LNCS*, pp. 253–273, 2011.
16. J. Chen, R. Gay, H. Wee. Improved Dual System ABE in Prime-Order Groups via Predicate Encodings. In *Eurocrypt 2015*, *LNCS*, pp. 595–624, 2015.
17. J. Chen, H. Wee. Fully, (Almost) Tightly Secure IBE from Standard Assumptions. In *Crypto 2013*, *LNCS*, pp. 435-460, 2013.
18. J. Chen, J. Gong, L. Kowalczyk, H. Wee. Unbounded ABE via Bilinear Entropy Expansion, Revisited. In *Eurocrypt 2018*, *LNCS*, pp. 503–534, 2018.
19. C. Delerablée. Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In *Asiacrypt 2007*, *LNCS*, pp. 200–215, 2007.
20. S. Gorbunov, V. Vaikuntanathan, H. Wee. Attribute-based encryption for circuits. In *STOC 2013*, pp. 545–554, 2013.
21. V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS 2006*, pp. 89–98, 2006.
22. M. Karchmer, A.Wigderson. On span programs. In Proc. of the Eighth Annual *Structure in Complexity Theory Conference*, IEEE, pp. 102–111, 1993.
23. A. Lewko, B. Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010*, *LNCS*, pp. 455–479, 2010.
24. A. Lewko, B. Waters. Decentralizing Attribute-Based Encryption In *Eurocrypt 2011*, *LNCS*, pp. 568-588, 2011.
25. A. Lewko, B. Waters. Unbounded HIBE and Attribute-Based Encryption In *Eurocrypt 2011*, *LNCS*, pp. 547–567, 2011.
26. A. Lewko, B. Waters. New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In *Crypto 2012*, *LNCS*, pp. 180–198.

27. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt 2010*, *LNCS*, pp. 62–91, 2010.
28. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS 2007*, pp. 195–203, 2007.
29. T. Okamoto, K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption.In *Crypto 2010*, *LNCS*, pp. 191–208, 2010.
30. T. Okamoto, K. Takashima, Fully Secure Unbounded Inner-Product and Attribute-Based encryption,. In *Asiacrypt 2012*, *LNCS*, pp. 349–366, 2012.
31. Y. Rouselakis, B. Waters Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM CCS 2013*, pp. 463–474, 2013.
32. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt 2005*, *LNCS*, pp. 457–473, 2005.
33. B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC 2011*, *LNCS*, pp. 53–70, 2011.
34. B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto 2009*, *LNCS*, pp. 619–636, 2009.
35. B. Waters. Functional Encryption for Regular Languages. In *Crypto 2012*, *LNCS*, pp. 218–235, 2012.
36. H. Wee. Dual System Encryption via Predicate Encodings. In *TCC 2014*, *LNCS*, pp. 616–637, 2014.
37. K. Yamada, N. Attrapadung, K. Emura, G. Hanaoka, K. Tanaka. Generic Constructions for Fully Secure Revocable Attribute-Based Encryption. In *ESORICS 2017 (2)*, *LNCS*, pp. 532–551, 2017.
38. S. Yamada, N. Attrapadung, G. Hanaoka, N. Kunihiro. A Framework and Compact Constructions for Non-monotonic Attribute-Based Encryption. In *PKC 2014*, *LNCS*, pp. 275–292, 2014.
39. D. Yang, B. Wang, X. Ban. Fully secure non-monotonic access structure CP-ABE scheme. In *KSII Trans. on Internet and Information Systems*, pp. 1315–1329, 2018.