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 [36], 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.,* [24,14,32,33,34,23]). 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 [36,24].

Previous Work on Predicate Compositions. We categorize as follows.

- **Static & Specific**. Dual-policy ABE (DP-ABE), introduced in [6], 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 [24,39].
- **Static & Small-class & Generic**. Attrapadung and Yamada [12] proposed a more general conversion that can combine ABE for any predicates that can be interpreted in the so-called *pair encoding* framework [7,8,1,2], but again, fixed for only the AND connector. A generic DUAL conversion, which swaps key and ciphertext attribute, was also proposed in [7,12]. 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 [42,19]. 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 $\mathcal{P} = \{P_1, \ldots, P_k\}$ 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) [25] 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 $M = (\mathbf{A}, (i_1, x_1), \ldots, (i_m, x_m))$ 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 $Y = \{(j_1, y_1), \ldots, (j_t, y_t)\}$. 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.

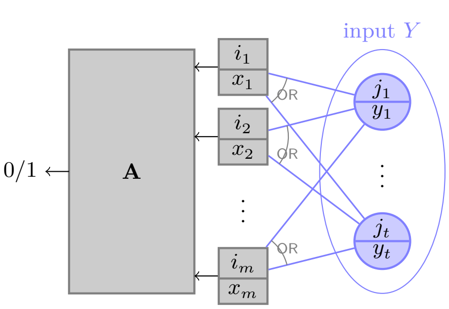


Fig. 1: Span program over predicates

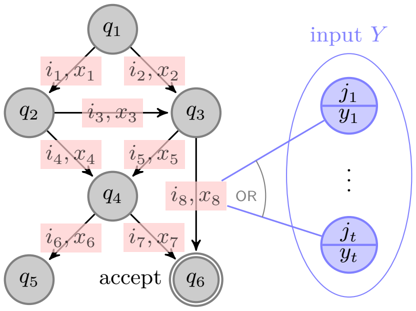


Fig. 2: Branching program over predicates

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 [8].
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 $\mathbf{y} = ((j_1, y_1), \ldots, (j_t, y_t))$ 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 [41,7], 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 C instead of 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

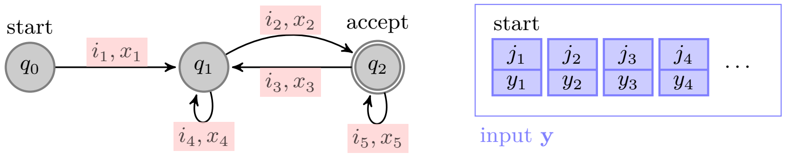


Fig. 3: DFA over predicates

for achieving fully secure ABE from a primitive called Pair Encoding Scheme (PES), proposed by Attrapadung [7]. PESs for many predicates have been proposed [7,12,8,2], notably, including regular language functionality [41,7]. 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 (discussed in §9 and §L), 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 [3,43], range/subset membership [10], regular string matching [41], 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 [32,11,44] or has some bounded attribute reuse [33,34]. 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 [35,7,2], even selectively secure one [35].
- 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 (see §L.1).

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 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 ABS use predicate encoding of [19]). 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 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.

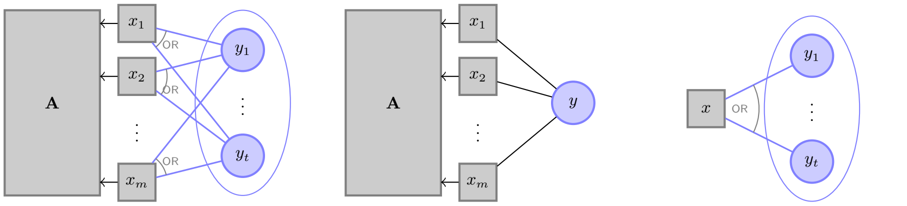
We provide some more related works and possible future directions in §L.6.

2 Intuition and Informal Overview

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

Pair Encoding. We first informally describe PES [7] as refined in [2]. It consists of two encoding algorithms as the main components. The ciphertext encoding 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 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 αs_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 αs_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 [20,1,2]. 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 αs_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, α 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 [15].

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 [40,27,30,42,7,2].

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.,* [29,35,7,21]. Rouselakis and Waters [35] were the first to (implicitly) use this exact encoding. Attrapadung [7] 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).

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

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 $\bar{\mathit{\Gamma}}_{\mathsf{IBBE}}$:

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

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:

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

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 α, 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, α, 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_{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 §L.5.)

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 [21]. Chen *et al.* [21] recently proposed unbounded ABE for MSP. Their approach conceptually converts a specific bounded scheme ([31]) 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 $[a, b] = \{ a, \ldots, b \}$. For $m \in \mathbb{N}$, let $[m] = \{ 1, \ldots, m \}$ and $[m]^+ = \{ 0, 1, \ldots, m \}$. 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 $\mathbf{a}[i, j] \coloneqq (\mathbf{a}[i], \mathbf{a}[i + 1], \ldots, \mathbf{a}[j])$. 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 span(S) as the set of all linear combinations of vectors in S. For polynomials $p = p(x_1, \ldots, x_n)$ and $g = g(y_1, \ldots, y_n)$, we denote a new polynomial $p|_{x_1 \mapsto g} \coloneqq p(g(y_1, \ldots, y_n), x_2, \ldots, x_n)$. 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 ℓ 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 κ 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 κ 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^λ and a parameter κ of predicate family P, and outputs a master public key PK and a master secret key MSK.
- $\mathsf{Encrypt}(y, M, \mathsf{PK}) \to \mathsf{CT}$: takes as input a ciphertext attribute $y \in \mathcal{Y}_\kappa$, a message $M \in \mathcal{M}$, and public key PK. It outputs a ciphertext CT. We assume that Y is implicit in 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 MSK. It outputs a secret key SK.
- $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK}) \to M$: given a ciphertext CT with its attribute y and the decryption key SK with its attribute x, it outputs a message M or $\perp$.

Correctness. Consider all parameters κ, 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. The standard notion for ABE is called full security. We omit it here and defer its definition to the §B.1, as we do not work directly on it but will rather infer the implication from pair encoding scheme (Proposition 2).

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

[5] It is also called public-index predicate encryption, classified in the definition of Functional Encryption [17]. 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 $y \in \mathcal{Y}_{(N,\mathsf{par})}$, EncCt outputs a vector of polynomial $\mathbf{c} = (c_1, \dots, c_{w_3})$ in *non-lone* variables $\mathbf{s} = (s_0, s_1, \dots, s_{w_1})$ and *lone* variables $\hat{\mathbf{s}} = (\hat{s}_1, \dots, \hat{s}_{w_2})$. 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 $x \in \mathcal{X}_{(N,\mathsf{par})}$, EncKey outputs a vector of polynomial $\mathbf{k} = (k_1, \dots, k_{m_3})$ in *non-lone* variables $\mathbf{r} = (r_1, \dots, r_{m_1})$ and *lone* variables $\hat{\mathbf{r}} = (\alpha, \hat{r}_1, \dots, \hat{r}_{m_2})$. 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, Pair outputs two matrices $\mathbf{E}, \overline{\mathbf{E}}$ of sizes $(w_1 + 1) \times m_3$ and $w_3 \times m_1$, respectively. ◊

Correctness. A PES is said to be correct if for every $\kappa = (N, \mathsf{par})$, $x \in \mathcal{X}_\kappa$ and $y \in \mathcal{Y}_\kappa$ such that $P_\kappa(x, y) = 1$, 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 Pair and correctness together at once is to show such a linear combination that evaluates to αs_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 α, 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 EncCt (ciphertext-encoding) and EncKey (key-encoding), respectively. We often omit writing w_1, w_2 and m_1, m_2 in the output of EncCt and 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 $\mathit{\Gamma} = (\mathsf{Param}, \mathsf{EncCt}, \mathsf{EncKey}, \mathsf{Pair})$ 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 EncB and EncR depends only on x, and 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 $\mathsf{Sym\text{-}Prop}^+$. This is w.l.o.g. since one can convert any PES with Sym-Prop to another with $\mathsf{Sym\text{-}Prop}^+$, with minimal cost. Such a conversion, which we denote as Plus-Trans, also appears in [2]; we recap it in §B.2.

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 [20] 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 §B.2 for self-containment.

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) [24]. We will mostly focus on *completely unbounded* variant [7,2], 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.,* [24,31].

[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 [29,34,21] 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*, $P^{\mathsf{KP\text{-}MSP}} = \{\, P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\} \,\}_\kappa$, is indexed by $\kappa = (N)$ and is defined as follows. Recall that $\mathbf{A}_{i:}$ denotes the i-th row of $\mathbf{A}$.

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

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 $\mathbf{1}_1^\ell \notin \mathrm{span}\{\, \mathbf{A}_{i:} \mid i \in Q \,\}$, 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 [24,13]. The procedure of embedding a boolean formula as a span program can be found in *e.g.,* §C of [28]. 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 [16,9]. We prove the implication for the case of PES in §B.3.

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)-$\mathsf{Sym\text{-}Prop}^+$ 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 α, b_1 do not appear elsewhere in $\mathbf{k}$.

(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]

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

[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 $\mathbf{c}$.

(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 $v \in [m_1]$. ♢

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 α appears only in k_1 and we match the monomial αs_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 α 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 Γ be a PES construction for P. We construct another PES Γ' for also the same P as follows. We denote this Γ' 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_{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_{new} be a new non-lone variable and α_{new} be the new special lone variable. Let $\tilde{\mathbf{k}}$ be exactly $\mathbf{k}$ but with α 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 Γ 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* Γ *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 §E.)

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.

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 $\mathbf{A}|_x := \{ \mathbf{A}_{i:} \mid P_\kappa(x, \pi(i)) = 1 \}$.

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

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

- $\mathsf{Param}'(\mathsf{par}) = \mathsf{Param}(\mathsf{par}) = n$. 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_{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
$$\mathbf{c}'^{(i)} := \mathbf{c}^{(i)}|_{b_1 s_0^{(i)} \mapsto \mathbf{A}_{i:} \mathbf{v}^\top}. \tag{12}$$

  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 $S := \{ i \in [m] \mid P_\kappa(x, \pi(i)) = 1 \}$. For $i \in S$, we can run $\mathsf{Pair}(x, \pi(i), N) \to (\mathbf{E}, \overline{\mathbf{E}})$. From the correctness of Γ, 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 Γ for P is (d_1, d_2)-admissible. Then,* $\mathsf{CP1\text{-}Trans}(\Gamma)$ *for* $\mathsf{CP1}[P]$ *is $(\ell + m(d_1 - 1), md_2)$-admissible, where $m \times \ell$ is the size of policy.*

Proof. We prove symbolic property of Γ' from that of Γ as follows.

Selective Symbolic Property. We define the following algorithms.

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

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

[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 $\tilde{\mathbf{s}}^{(i)} = (s_1^{(i)}, \ldots, s_{w_{1,i}}^{(i)})$.

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 ℓ 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

$$\mathbf{B}'_j = \left(\begin{array}{cccc} \mathbf{e}_j^{(1)}\mathbf{A}_{1,1} & & \cdots & \mathbf{e}_j^{(m)}\mathbf{A}_{m,1} \\ \vdots & & & \vdots \\ \mathbf{e}_j^{(1)}\mathbf{A}_{1,\ell} & & \cdots & \mathbf{e}_j^{(m)}\mathbf{A}_{m,\ell} \\ \hline \tilde{\mathbf{B}}_j^{(1)} & & & \\ & \tilde{\mathbf{B}}_j^{(2)} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{B}}_j^{(m)} \end{array}\right) \in \mathbb{Z}_N^{d'_1 \times d'_2}, \tag{13}$$

$$\mathbf{s}'^{(i)}_t = (0, \ldots, 0,\ \overset{\substack{\text{block } i \\ \downarrow}}{\mathbf{s}_t^{(i)}}\ , 0, \ldots, 0) \in \mathbb{Z}_N^{1 \times d'_2},$$

$$\hat{\mathbf{s}}'^{(i)}_z = \big(\hat{\mathbf{s}}_z^{(i)}[1]\mathbf{A}_{i:},\ 0, \ldots, 0, \overset{\substack{\text{block } i \\ \downarrow}}{\hat{\mathbf{s}}_z^{(i)}[2, d_1]}, 0, \ldots, 0\big) \in \mathbb{Z}_N^{1 \times d'_1}, \tag{14}$$

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

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

$\boxed{\mathsf{EncR}'(x, (\mathbf{A}, \pi))}$: 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 $\mathbf{a} = \mathbf{1}_1^{d_2} \in \mathbb{Z}_N^{1 \times d_2}$.
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
$$\mathbf{r}'_v = -\Big(\boldsymbol{\omega},\ g_1\mathbf{r}_v^{(1)}[2, d_1], \ldots, g_m\mathbf{r}_v^{(m)}[2, d_1]\Big) \in \mathbb{Z}_N^{1 \times d'_1}, \tag{15}$$
$$\hat{\mathbf{r}}'_u = -(g_1\hat{\mathbf{r}}_u^{(1)}, \ldots, g_m\hat{\mathbf{r}}_u^{(m)}) \in \mathbb{Z}_N^{1 \times d'_2}. \tag{16}$$
5. Output $(\mathbf{r}'_1, \ldots, \mathbf{r}'_{m_1};\ \mathbf{a}_{\text{new}}, \hat{\mathbf{r}}'_1, \ldots, \hat{\mathbf{r}}'_{m_2})$.

[10] That is, the i-th block of a vector $\mathbf{h} \in \mathbb{Z}_N^{1 \times d'_1}$ is $\mathbf{h}[\ell + (d_1 - 1)(i - 1) + 1, \ell + (d_1 - 1)i]$.

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

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

$$\sum_{z\in[w_{2,i}]} \eta^{(i)}_{p,z}\hat{s}'^{(i)}_z + \eta^{(i)}_{p,0,1}(\mathbf{A}_{i,1}b_1 s_{\mathrm{new}} + \sum_{\iota=2}^{\ell}\mathbf{A}_{i,\iota}v_\iota) + \sum_{\substack{t\in[w_{1,i}]\\ j\in[2,n]}} \eta^{(i)}_{p,t,j}b_j s'^{(i)}_t. \tag{17}$$

Substituting $\hat{s}'^{(i)}_z : (\hat{\mathbf{s}}'^{(i)}_z)^\top$, $b_1 s_{\mathrm{new}} : \mathbf{B}'_1(\mathbf{s}_{\mathrm{new}})^\top$, $v_\iota : (\mathbf{v}'_\iota)^\top$, $b_j s'^{(i)}_t : \mathbf{B}'_j(\mathbf{s}'^{(i)}_t)^\top$, into $c'^{(i)}_p$ 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, Γ, which ensures that the substitution of $c^{(i)}_p$ 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 $k_p = \sum_{u\in[m_2]} \phi_{p,u}\hat{r}_u + \sum_{v\in[m_1],j\in[2,n]} \phi_{p,v,j} r_v b_j$. Substituting $\hat{r}_u : \hat{\mathbf{r}}'_u$, $r_v b_j : \mathbf{r}'_v\mathbf{B}'_j$ 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 $\mathsf{EncS}(x, \pi(i)) \to \left(\mathbf{s}^{(i)}_0, \ldots, \mathbf{s}^{(i)}_{w_{1,i}};\ \hat{\mathbf{s}}^{(i)}_1, \ldots, \hat{\mathbf{s}}^{(i)}_{w_{2,i}}\right)$, where $\mathbf{s}^{(i)}_t \in \mathbb{Z}_N^{1\times d_2}$, and $\hat{\mathbf{s}}^{(i)}_z \in \mathbb{Z}_N^{1\times d_1}$.
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 $\mathbf{s}_{\mathrm{new}} = \mathbf{1}^{d_2}_1$, $\mathbf{s}'^{(i)}_t = q_i\mathbf{s}^{(i)}_t$, $\hat{\mathbf{s}}'^{(i)}_z = q_i\hat{\mathbf{s}}^{(i)}_z$, and $\mathbf{v}'_\iota = \omega_\iota\mathbf{1}^{d_1}_1$, 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 $\mathbf{a}_{\mathrm{new}}\mathbf{s}^\top_{\mathrm{new}} = \mathbf{1}^{d_2}_1(\mathbf{1}^{d_2}_1)^\top = 1 \neq 0$, 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 Γ. It remains to consider the substitution for $\mathsf{ct\text{-}enc}$ $\mathbf{c}'$. For $i \in [m]$, $p \in [w_{3,i}]$, the polynomial $c^{(i)}_p$ is depicted in Eq. (17). We have that the middle sum term $\mathbf{A}_{i:}\mathbf{v}^\top$ is substituted and evaluated to $q_i(\mathbf{1}^{d_2}_1)^\top$. Let $\mathbf{u}^\top_i \in \mathbb{Z}_N^{d_1\times 1}$ 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 $\mathbf{s}_t'^{(i)}$ and $\hat{\mathbf{s}}_z'^{(i)}$, it is straightforward to see that the substitution for $c_p'^{(i)}$ (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 Γ. □

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: $\mathbf{r}_v'\mathbf{B}_j' = -\big(g_1 \mathbf{r}_v^{(1)}\mathbf{B}_j^{(1)}, \ldots, g_m \mathbf{r}_v^{(m)}\mathbf{B}_j^{(m)}\big)$ (*cf.* Eq. (27) in §F), 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 [12,2]—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 Γ be a PES construction for a P satisfying admissibility. We construct a PES Γ' for $\mathsf{KP1}[P]$ as follows. Denote this Γ' by $\mathsf{KP1\text{-}Trans}(\Gamma)$.

- $\mathsf{Param}'(\mathsf{par}) = \mathsf{Param}(\mathsf{par}) = n$. 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 $\mathbf{s}\mathbf{E}(\mathbf{k}'^{(i)})^\top + \mathbf{c}\overline{\mathbf{E}}(\mathbf{r}^{(i)})^\top = \alpha^{(i)} s_0 = \mathbf{A}_{i:}\mathbf{v}^\top s_0$. 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 Γ for P is (d_1, d_2)-admissible. Then, the the PES* $\mathsf{KP1\text{-}Trans}(\Gamma)$ *for* $\mathsf{KP1}[P]$ *satisfies* $(md_1,\, m'd_2)$-$\mathsf{Sym\text{-}Prop}^+$*, where $m \times \ell$ is the size of policy and $m' = \max\{m, \ell\}$.* [11]

The proof is analogous to $\mathsf{CP1\text{-}Trans}$, and is deferred to G. Note that, unlike $\mathsf{CP1\text{-}Trans}$, $\mathsf{KP1\text{-}Trans}$ 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

- $\bar{\mathcal{X}}_\kappa = \{\, (\mathbf{A}, \pi) \mid \mathbf{A} \in \mathbb{M}(\mathbb{Z}_N),\ \pi : [m] \to \mathbb{X}_\kappa \,\}$.
- $\bar{\mathcal{Y}}_\kappa = 2^{\mathbb{Y}_\kappa}$.
- $\bar{P}_\kappa((\mathbf{A}, \pi), Y) = 1 \iff \mathbf{1}_1^\ell \in \text{span}(\mathbf{A}|_Y)$, where[12]

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

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

[11] As noted in Remark 4, we have in particular that m, ℓ will affect only the size of $\mathsf{q\text{-}ratio}$ assumption, but do not pose any bound on the policy size for the ABE scheme.

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

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 (see Definition 12 in §G for a detailed description), 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 $\mathsf{DS}[\mathcal{P}] = \left\{ \bar{P}_\kappa \right\}_\kappa$ where we let the predicate be $\bar{P}_\kappa : \mathbb{X}_\kappa \times \mathbb{Y}_\kappa \to \{0,1\}$ with

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

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

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 §H.

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 $\mathsf{CP1\text{-}Trans}$ and $\mathsf{KP1\text{-}Trans}$ to a PES for $\mathsf{DS}[\mathcal{P}]$ to obtain a PES for $\mathsf{KP}[\mathcal{P}]$. Note that we apply $\mathsf{Layer\text{-}Trans}$ 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 $\mathcal{P}$.

7.1 Simple Direct Sum by Parameter Concatenation

Construction 4. Let $\Gamma^{(j)}$ be a PES for $P^{(j)}$. Also let $\mathbf{\Gamma} = (\Gamma^{(1)}, \ldots, \Gamma^{(k)})$. We construct a PES Γ' for $\mathsf{DS}[\mathcal{P}]$, where $\mathcal{P} = \{P^{(1)}, \ldots, P^{(k)}\}$, as follows. For further use, we denote this Γ' 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 $\mathbf{b}^{(j)} = (b^{(j)}_1, \ldots, b^{(j)}_{n_j})$. 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 $\mathsf{Concat\text{-}Trans}(\boldsymbol{\Gamma})$, 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 §H. 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 Γ 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 $x \in \mathcal{X}_\kappa$. $\diamondsuit$

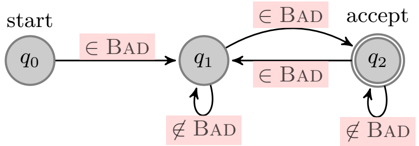
Lemma 3. *Suppose that, for all $j \in [k]$, the PES $\Gamma^{(j)}$ for predicate family $P^{(j)}$ satisfies (d_1,d_2)-$\mathsf{Sym\text{-}Prop}^{++}$. Then, the PES $\mathsf{Concat\text{-}Trans}(\boldsymbol{\Gamma})$ for predicate family $\mathsf{DS}[\mathcal{P}]$, where $\mathcal{P} = \{P^{(1)},\dots,P^{(k)}\}$,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

7.2 Efficient Direct Sum with Parameter Reuse

Construction 5. Let $\Gamma^{(j)}$ be a PES for $P^{(j)}$. Also let $\boldsymbol{\Gamma} = (\Gamma^{(1)},\dots,\Gamma^{(k)})$. We construct a PES Γ' for $\mathsf{DS}[\mathcal{P}]$, where $\mathcal{P} = \{P^{(1)},\dots,P^{(k)}\}$, as follows. We denote this scheme by $\mathsf{Reuse\text{-}Trans}(\boldsymbol{\Gamma})$. The intuition is to use two new parameters g_j, h_j specific to $\Gamma^{(j)}$, 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 $\mathbf{b} = (b_1,\dots,b_n,g_1,\dots,g_k,h_1,\dots,h_k)$. Also denote $\mathbf{b}_j = (b_1,\dots,b_{n_j})$.
- $\mathsf{EncCt}'((j,y),N)$. Run $\mathsf{EncCt}^{(j)}(y,N) \to \mathbf{c} = \mathbf{c}(\mathbf{s},\hat{\mathbf{s}},\mathbf{b}_j)$. Let s_{new} be the new special non-lone variable. Output $\mathbf{c}' = \big(\mathbf{c},\;\; g_j s_0 + h_j s_{\text{new}}\big)$.
- $\mathsf{EncKey}'((i,x),N)$. Run $\mathsf{EncKey}^{(i)}(x,N) \to \mathbf{k} = \mathbf{k}(\mathbf{r},\hat{\mathbf{r}},\mathbf{b}_i)$. Let r_{new} be a new non-lone variable and α_{new} be the new special lone variable. Let $\tilde{\mathbf{k}}$ be exactly $\mathbf{k}$ but with α being replaced by $r_{\text{new}} g_i$. Output $\mathbf{k}' = \big(\tilde{\mathbf{k}},\;\; \alpha_{\text{new}} + r_{\text{new}} h_i\big)$.

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 $\mathbf{s}\mathbf{E}\mathbf{k}^\top + \mathbf{c}\overline{\mathbf{E}}\mathbf{r}^\top = \alpha s_0 = (r_{\text{new}} g_j) s_0$. Hence, we have the following, as required: $\big(\,\alpha_{\text{new}} + r_{\text{new}} h_j\,\big) s_{\text{new}} - r_{\text{new}}\big(\,g_j s_0 + h_j s_{\text{new}}\,\big) + (r_{\text{new}} g_j) s_0 = \alpha_{\text{new}} s_{\text{new}}$.



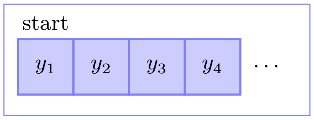


Fig. 5: Predicative DFA for language of sentences that start with a bad word and have an even number of the total bad words. Based predicates for testing membership/non-membership can use IBBE, IBR, defined in §9.2, respectively.

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 $\mathcal{P} = \{P^{(1)}, \ldots, P^{(k)}\}$, satisfies (d_1, d_2)-$\mathsf{Sym\text{-}Prop}^+$.* (The proof is deferred to §H.2.)

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_κ 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 $Y = (y_1, \ldots, y_\ell) \in (\mathcal{Y}_\kappa)^*$, where ℓ is unbounded. A predicative automata $M = (Q = \{q_0, \ldots, q_{\sigma-1}\}, \mathcal{T}, q_0, q_{\sigma-1})$ accepts Y if there exists a sequence of states $(q^{(1)}, \ldots, q^{(\ell)}) \in Q^\ell$ 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) [41,7,2], 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 $\mathsf{KA1}[P] = \{ \bar{P}_\kappa \}_\kappa$ where $\bar{P}_\kappa : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{ 0, 1 \}$ by letting

- $\bar{\mathcal{X}}_\kappa = \{ M \mid M \text{ is a predicative automata over } P_\kappa \ \}$.

- $\bar{\mathcal{Y}}_\kappa = (\mathcal{Y}_\kappa)^*$.
- $\bar{P}_\kappa(M, Y) = 1 \iff M$ 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 [7], 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 [7], 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 Γ be a PES construction for P. We construct a PES Γ' for $\mathsf{KA1}[P]$ as follows. For further use, we denote this Γ' 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:

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

  and relate these two sets of variables via:

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

  We then define $c_0' \coloneqq h_0 s_{\text{new}}^{(0)}$ and, for $i \in [\ell]$,

  $$c_i' \coloneqq h_1 s_{\text{new}}^{(i-1)} + g_1 s_0'^{(i-1)} + h_2 s_{\text{new}}^{(i)} + g_2 s_0'^{(i)},$$

  where $s_{\text{new}}^{(0)}, \ldots, s_{\text{new}}^{(\ell)}$ are new non-lone variables with $s_{\text{new}}^{(\ell)}$ being special. Finally, it outputs $\mathbf{c}' \coloneqq \big(c_0', c_1', \ldots, c_\ell', \, (\mathbf{c}^{(1,i)}, \mathbf{c}^{(2,i)})_{i \in [\ell]} \big)$.

- $\mathsf{EncKey}'(M, N)$. Parse $M = (Q, \mathcal{T}, q_0, q_{\sigma-1})$ and parse $\mathcal{T} = \{\, (q_{\upsilon_t}, q_{\omega_t}, x_t) \,\}_{t\in[m]}$ where each $\upsilon_t, \omega_t \in [0, \sigma-1]$. [13] Let $u_0, u_1, \ldots, u_{\sigma-1}$ 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.

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

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

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

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 $(q^{(1)}, \ldots, q^{(\ell)}) \in Q^\ell$ 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

$$\begin{aligned} D_{1,i} &:= \alpha^{(1,t_i)} s_0^{(1,i)} = \big(r^{(t_i)}_{\text{new}} g_1\big) s_0'^{(i-1)}, \\ D_{2,i} &:= \alpha^{(2,t_i)} s_0^{(2,i)} = \big(r^{(t_i)}_{\text{new}} g_2\big) s_0'^{(i)}. \end{aligned}$$

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

Theorem 3. *Suppose a PES Γ 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 ℓ 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 §I. 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 $\mathsf{KP1}$ case, the proof is somewhat more complex.

[13] υ_t, ω_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 also offer more discussions in §L.3, §L.4, 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_σ 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 Γ_y be an induced subgraph of Γ that contains exactly all the edges e such that $P_\kappa(L(e), y) = 1$. Such a PBP M accepts y if Γ_y contains a directed path from the start node, q_σ, 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 [8]. For self-containment, we provide this in §J.

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 ℓ, it includes all the possible nesting structures that have at most ℓ 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 ℓ, 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 ℓ 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 ℓ-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 §K.

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 [7,2]. 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 $P^{\mathsf{IBBE}}(x,Y) = 1 \Leftrightarrow x \in Y$.[14]
  - It is clear that P^{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^{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 $P^{\mathsf{TIBBE}}((i,x),Y) = 1 \Leftrightarrow (i = 1 \wedge x \in Y) \vee (i = 2 \wedge x \notin Y)$.[15]
  - It is clear that P^{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 [7] 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 also explicitly describe this induced definition in §D.

For self-containment, we provide PES constructions for P^{IBE} and P^{NIBE} in §C.

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^{IBE} and P^{NIBE}. An explicit description of our PES for it is given in §D. We have to rely on the q-ratio assumption, inherited from the framework of [2][16]; nevertheless, all the current *completely unbounded* ABE for even *monotone* span programs still also need q-type assumptions [35,7,2], even *selectively* secure one [35]. 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 §L.1.

For the CP-ABE case, a fully secure completely-unbounded scheme for NSP was recently and independently reported in [45]. Their scheme is constructed in composite-order groups. Our instantiated CP-ABE for NSP is in prime-order groups, and unlike [45] of which proof is complex and specific, ours can be obtained in a modular manner. Table 2 for CP-ABE for NSP is deferred to §A.

[14] IBBE is for ID-based broadcast encryption [22]; IBR is for ID-based revocation [11].

[15] This is a unified notion for IBBE and IBR, and is called two-mode IBBE in [44].

[16] In defense, we also provide a positive remark towards the q-ratio assumption in §L.2.

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

Schemes		$\|\mathsf{PK}\|$	$\|\mathsf{SK}\|$	$\|\mathsf{CT}\|$	Unbounded			Security	Assumption
					|policy|/	multi-use/	|attrib. set|		
OSW07 [32]	I	$O(T)$	$O(m)$	$O(T)$	✓	✓		selective	DBDH
	II	$O(T)$	$O(m\log(T))$	$O(t)$	✓	✓		selective	DBDH
OT10 [33]		$O(TR)$	$O(m)$	$O(tR)$	✓			full	DLIN
OT12 [34]		$O(1)$	$O(m)$	$O(tR)$	✓		✓	full	DLIN
ALP11 [11]		$O(T)$	$O(Tm)$	$O(1)$	✓	✓		selective	T-DBDHE†
YAHK14 [44]	I	$O(T)$	$O(Tm)$	$O(1)$	✓	✓		selective	T-DBDHE†
	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†
Our KP-NSP	I	$O(1)$	$O(m)$	$O(t)$	✓	✓	✓	full	qratio†
	II	$O(T^2)$	$O(T^3m)$	$O(1)$	✓	✓		full	qratio†
	III	$O(M^2+ML)$	$O(1)$	$O(t(M^3+M^2L))$		✓	✓	full	qratio†

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, ℓ, respectively (if required). Assumptions with † are q-type assumptions.

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 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 1,2. We derive their complexities in §D.

Revisiting the Okamoto-Takashima Definition. The Okamoto-Takashima type ABE [33,34] 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 §D.

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. M. Ambrona, G. Barthe, R. Gay, H. Wee. Attribute-Based Encryption in the Generic Group Model: Automated Proofs and New Constructions. In *ACM CCS'17*, pp. 647–664, 2017.
5. B. Applebaum, B. Arkis, P. Raykov, P. N. Vasudevan. Conditional Disclosure of Secrets: Amplification, Closure, Amortization, Lower-Bounds, and Separations. In *Crypto (1) 2017*, *LNCS*, pp.727–757, 2017.
6. N. Attrapadung, H. Imai. Dual-Policy Attribute Based Encryption. In *ACNS 2009*, *LNCS*, pp. 168–185, 2009.

7. 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.
8. N. Attrapadung. Dual System Encryption Framework in Prime-Order Groups via Computational Pair Encodings. In *Asiacrypt 2016*, *LNCS*, pp. 591–623, 2016.
9. 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.
10. 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.
11. N. Attrapadung, B. Libert, E. Panafieu. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In *PKC 2011*, *LNCS*, pp. 90–108.
12. 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.
13. A. Beimel. Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
14. J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. In *IEEE S&P 2007*, pp. 321–334, 2007.
15. 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.
16. D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt 2008*, *LNCS*, pp. 455–470, 2008.
17. D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC 2011*, *LNCS*, pp. 253–273, 2011.
18. J.-H. Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In *Eurocrypt 2006*, *LNCS* 4004, pp. 1–11, 2006.
19. 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.
20. J. Chen, H. Wee. Fully, (Almost) Tightly Secure IBE from Standard Assumptions. In *Crypto 2013*, *LNCS*, pp. 435-460, 2013.
21. J. Chen, J. Gong, L. Kowalczyk, H. Wee. Unbounded ABE via Bilinear Entropy Expansion, Revisited. In *Eurocrypt 2018*, *LNCS*, pp. 503–534, 2018.
22. C. Delerablée. Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In *Asiacrypt 2007*, *LNCS*, pp. 200–215, 2007.
23. S. Gorbunov, V. Vaikuntanathan, H. Wee. Attribute-based encryption for circuits. In *STOC 2013*, pp. 545–554, 2013.
24. 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.
25. M. Karchmer, A.Wigderson. On span programs. In Proc. of the Eighth Annual *Structure in Complexity Theory Conference*, IEEE, pp. 102–111, 1993.
26. L. Kowalczyk, J. Liu, T. Malkin, K. Meiyappan. Mitigating the One-Use Restriction in Attribute-Based Encryption. In *ICISC 2018*, *LNCS*, pp. 23–36, 2018.
27. 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.
28. A. Lewko, B. Waters. Decentralizing Attribute-Based Encryption In *Eurocrypt 2011*, *LNCS*, pp. 568-588, 2011.
29. A. Lewko, B. Waters. Unbounded HIBE and Attribute-Based Encryption In *Eurocrypt 2011*, *LNCS*, pp. 547–567, 2011.

30. A. Lewko, B. Waters. New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In *Crypto 2012*, *LNCS*, pp. 180–198.
31. 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.
32. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS 2007*, pp. 195–203, 2007.
33. T. Okamoto, K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption.In *Crypto 2010*, *LNCS*, pp. 191–208, 2010.
34. T. Okamoto, K. Takashima, Fully Secure Unbounded Inner-Product and Attribute-Based encryption,. In *Asiacrypt 2012*, *LNCS*, pp. 349–366, 2012.
35. Y. Rouselakis, B. Waters Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM CCS 2013*, pp. 463–474, 2013.
36. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt 2005*, *LNCS*, pp. 457–473, 2005.
37. N. Sullivan. Geo manager. In *Real World Crypto'18*. Available at YouTube.
38. K. Takashima. New Proof Techniques for DLIN-Based Adaptively Secure Attribute-Based Encryption. In *ACISP 2017*, *LNCS*, pp.85–105, 2017
39. B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC 2011*, *LNCS*, pp. 53–70, 2011.
40. B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto 2009*, *LNCS*, pp. 619–636, 2009.
41. B. Waters. Functional Encryption for Regular Languages. In *Crypto 2012*, *LNCS*, pp. 218–235, 2012.
42. H. Wee. Dual System Encryption via Predicate Encodings. In *TCC 2014*, *LNCS*, pp. 616–637, 2014.
43. 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.
44. 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.
45. 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.

Supplementary Materials

A Supplementary Table and Figure

This section provides a summary table for CP-ABE, for supplementary to §9.2.

Table 2: Summary for CP-ABE for non-monotone span programs with large universe.

Schemes	$\lvert\mathsf{PK}\rvert$	$\lvert\mathsf{SK}\rvert$	$\lvert\mathsf{CT}\rvert$	Unbounded			Security	Assumption
				|policy|	/multi-use/	|attrib. set|		
OT10 [33]	$O(TR)$	$O(tR)$	$O(m)$	✓			full	DLIN
OT12 [34]	$O(1)$	$O(tR)$	$O(m)$	✓		✓	full	DLIN
YAHK14 [44]	$O(1)$	$O(t)$	$O(m)$	✓	✓	✓	selective	m-B[†]
YWB18 [45]	$O(1)$	$O(t)$	$O(m)$	✓	✓	✓	full	q-PBDHE[†] (Composite-order)
Our CP-NSP I	$O(1)$	$O(t)$	$O(m)$	✓	✓	✓	full	qratio[†]
II	$O(T^2)$	$O(1)$	$O(T^3m)$	✓	✓		full	qratio[†]
III	$O(M^2+ML)$	$O(t(M^3+M^2L))$	$O(1)$		✓	✓	full	qratio[†]

Note: $t = \lvert$attribute set$\rvert$, $m \times \ell$ is the span program size, R is the attribute multi-use bound, T, M, L are the maximum bound for t, m, ℓ, respectively (if required). Note that only the scheme of [45] is in composite-order groups. Assumptions with [†] are q-type assumptions.

B Recapped Known Definitions and Results

B.1 Security Definition for ABE

For self-containment, here we describe the security definition for ABE, deferred from §3.1. An ABE scheme for predicate family P is fully secure if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ has non-negligible advantage in the following game between $\mathcal{A}$ and the challenger $\mathcal{C}$.

1. **Setup**: $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, \kappa) \to (\mathsf{PK}, \mathsf{MSK})$ and hands PK to $\mathcal{A}$.
2. **Phase 1**: $\mathcal{A}$ makes a j-th private key query for $X_j \in \mathcal{X}_\kappa$. $\mathcal{C}$ returns SK_j by computing $\mathsf{SK}_j \leftarrow \mathsf{KeyGen}(X_j, \mathsf{MSK}, \mathsf{PK})$.
3. **Challenge**: $\mathcal{A}$ submits equal-length messages M_0, M_1 and a target ciphertext attribute $y^\star \in \mathcal{Y}_\kappa$ with the restriction that $P_\kappa(x_j, y^\star) = 0$ for all $j \in [1, q_1]$. $\mathcal{C}$ flips a bit $b \xleftarrow{\$} \{0,1\}$ and returns $\mathsf{CT}^\star \leftarrow \mathsf{Encrypt}(y^\star, M_b, \mathsf{PK})$.
4. **Phase 2**: $\mathcal{A}$ continues to make a j-th private key query for $X_j \in \mathcal{X}_\kappa$ under the restriction $P_\kappa(x_j, y^\star) = 0$. $\mathcal{C}$ returns $\mathsf{SK}_j \leftarrow \mathsf{KeyGen}(X_j, \mathsf{MSK}, \mathsf{PK})$.
5. **Guess**: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and wins if $b' = b$. The advantage of $\mathcal{A}$ is defined as $|\Pr[b = b'] - \frac{1}{2}|$.

B.2 Main Theorem in the Agrawal-Chase Framework

For self-containment, we recap the main theorem in the Agrawal-Chase framework [2], deferred from the discussion in §3.3.

Proposition 2 ([2]). *Suppose there exists a PES that satisfies* (d_1, d_2)-$\mathsf{Sym\text{-}Prop}$ *for a predicate family* P*. Then, there exists a fully secure ABE scheme for the predicate family* P *in prime-order bilinear maps under the* (D_1, D_2)-$\mathsf{q\text{-}ratio}$ *and the* k*-linear assumptions for any* $k \geq 2$*, where* $D_1 = \mathsf{max}(d_1, d_2 - 1) + M_1 + 1$ *and* $D_2 = d_2 + W_1 + 1$*, and* M_1 *and* W_1 *are the maximums of the number of* key-enc *and* ct-enc *non-lone variables, respectively, in the encoding among respective key and ciphertext queries (in the security game for ABE).*

Remark 2 (Enhanced Symbolic Property [2]). A PES satisfying (d_1, d_2)-Sym-Prop is indeed not *directly* sufficient for constructing fully secure ABE. Agrawal and Chase [2] provide a more constrained definition of Sym-Prop called Enhanced Symbolic Property, or Sym-Prop*. An ABE scheme constructed via their generic construction over a PES with Sym-Prop* is proved fully secure. As shown in [2], most previous concrete PESs [7,2] already satisfy Sym-Prop*. Otherwise, Agrawal and Chase [2] also provide a generic conversion which converts any PES with (d_1, d_2)-Sym-Prop to a PES with (d_1, d_2)-Sym-Prop*.

Remark 3. As noted in [2], if a scheme satisfies (d_1, d_2)-Sym-Prop, then it also satisfies (d'_1, d'_2)-Sym-Prop for any $d'_1 \geq d_1$ and $d'_2 \geq d_2$.

Remark 4. As noted in [2], the parameter (d_1, d_2) for symbolic property of a PES will affect only the size of q-ratio assumption, but will not pose any bound on the ABE scheme syntax, constructed over the PES.

Constraint. We will use a constraint that $\mathbf{a} = \mathbf{1}_1^{d_2}$. This constraint is implied from Sym-Prop* (but we do not need all the other constraints in Sym-Prop*, which require a bit more). Therefore, any PES with Sym-Prop can be converted to a PES with Sym-Prop$^+$ using the conversion in [2].[17] We describe this conversion below.

Construction 7. Let Γ be a PES construction for P. We construct another PES Γ' for also the same P as follows. We denote this Γ' by $\mathsf{Plus\text{-}Trans}(\Gamma)$.

- $\mathsf{Param}'(\mathsf{par})$. If $\mathsf{Param}(\mathsf{par})$ returns n, then output $n+1$. Denote $\mathbf{b} = (b_1, \ldots, b_n)$ and $\mathbf{b}' = (\mathbf{b}, f)$.
- $\mathsf{EncCt}'(y, N)$. Run $\mathsf{EncCt}(y, N) \to \mathbf{c}$. Output $\mathbf{c}' = (\mathbf{c},\ fs_0)$.
- $\mathsf{EncKey}'(x, N)$. Run $\mathsf{EncKey}(x, N) \to \mathbf{k}$. Let r_{new} be a new non-lone variable. Let $\tilde{\mathbf{k}}$ be exactly $\mathbf{k}$ but with α being replaced by $\alpha + r_{\text{new}} f$. Output $\tilde{\mathbf{k}}$.

Proposition 3 ([2]). *Suppose that* Γ *for* P *satisfies* (d_1, d_2)-Sym-Prop*. Then,* $\mathsf{Plus\text{-}Trans}(\Gamma)$ *for* P *satisfies* (d_1, d_2)-Sym-Prop$^+$*.*

[17] In fact, this requires only the first conversion out of the three consecutive conversions in [2] that turns a PES with Sym-Prop to a PES with Sym-Prop*.

B.3 Embedding Lemma

For arguing implications among PESs, we use the embedding lemma, deferred from §3.4. Such a lemma is already known and applied for arguing implications among ABE schemes [16,9]. We adapt to the case of PES here.

Definition 10. Let $P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\}$, and $P'_{\kappa'} : \mathcal{X}'_{\kappa'} \times \mathcal{Y}'_{\kappa'} \to \{0,1\}$ be two predicate families, indexed by $\kappa = (N, \mathsf{par}) \in \mathcal{K}$ and $\kappa' = (N, \mathsf{par}') \in \mathcal{K}'$, respectively. We say that P' *can be embedded into* P if there exists three efficient mappings $f_\mathsf{p}, f_\mathsf{e}, f_\mathsf{k}$ where $f_\mathsf{p} : \mathcal{K}' \to \mathcal{K}$ maps $\kappa' = (N, \mathsf{par}') \mapsto \kappa = (N, \mathsf{par})$ and $f_\mathsf{e} : \mathcal{X}'_{\kappa'} \to \mathcal{X}_\kappa$, $f_\mathsf{k} : \mathcal{Y}'_{\kappa'} \to \mathcal{Y}_\kappa$ such that for all $x' \in \mathcal{X}'_{\kappa'}, y' \in \mathcal{Y}'_{\kappa'}$, we have:

$$P'_{\kappa'}(x', y') = 1 \iff P_\kappa(f_\mathsf{e}(x'), f_\mathsf{k}(y')) = 1. \tag{19}$$

Lemma 5. *If P' can be embedded into P, then any symbolic-secure PES for P implies a symbolic-secure PES for P'.*

Proof sketch. Let Γ be a PES for P. We construct a PES Γ' for P' by simply defining $\mathsf{Param}'(\mathsf{par}') = \mathsf{Param}(f_\mathsf{p}(\mathsf{par}'))$, $\mathsf{EncCt}'(y', N) = \mathsf{EncCt}(f_\mathsf{e}(y'), N)$, $\mathsf{EncKey}'(x', N) = \mathsf{EncCt}(f_\mathsf{k}(x'), N)$, and $\mathsf{Pair}'(x', y', N) = \mathsf{Pair}(f_\mathsf{k}(x'), f_\mathsf{e}(y'), N)$. The correctness and security is guaranteed by the forward and backward direction of Eq. (19), respectively. □

C Basic Concrete Pair Encoding Schemes

In this section, we describe pair encoding schemes for some basic predicates.

C.1 Basic Schemes

Construction 8. A PES for P^{IBE}. The encoding is that of the Boneh-Boyen scheme [15] (which is also used in [30,7]).

- $\mathsf{Param} \to 2$. Denote $\mathbf{b} = (b_1, b_2)$.
- $\mathsf{EncCt}(y, N) \to c_1 = (yb_1 + b_2)s_0$.
- $\mathsf{EncKey}(x, N) \to k_1 = \alpha + r_1(xb_1 + b_2)$.
- Pair. Suppose $x = y$. We have $k_1 s_0 - r_1 c_1 = \alpha s_0$.

Selective Symbolic Property of Construction 8.

- $\mathsf{EncBS}(y)$ outputs $(b_1 : -1,\ b_2 : y,\ s_0 : 1)$. Note that $d_1 = d_2 = 1$.
- $\mathsf{EncR}(x, y)$, where $x \neq y$, outputs $(r_1 : \frac{1}{x-y},\ \alpha : 1)$.
- We can verify that
  - $\alpha s_0 : (1)(1) = 1 \neq 0$.
  - $c_1 : (y(-1) + y)(1) = 0$
  - $k_1 : 1 + \frac{1}{x-y}(x(-1) + y) = 0$.

Co-selective Symbolic Property of Construction 8.

- $\mathsf{EncBR}(x)$ outputs $(b_1 : (0,-1),\ b_2 : (-1,x),\ r_1 : 1,\ \alpha : (1,0))$. Note that $d_1 = 1, d_2 = 2$.
- $\mathsf{EncS}(x,y)$, where $x \neq y$, outputs $(s_0 : (1, \frac{1}{x-y}))$.
- We can verify that
  - $\alpha s_0 : (1,0) \cdot (1, \frac{1}{x-y})^\top = 1 \neq 0$.
  - $k_1 : (1,0) + (1)(x(0,-1) + (-1,x)) = (0,0)$.
  - $c_1 : (y(0,-1) + (-1,x)) \cdot (1, \frac{1}{x-y})^\top = 0$.

Construction 9. A PES for P^{NIBE} (the negated predicate of IBE). This is extracted (and simplified) as a special case of negated spatial encryption of [7].

- $\mathsf{Param} \to 2$. Denote $\mathbf{b} = (b_1, b_2)$.
- $\mathsf{EncCt}(y,N) \to c_1 = (yb_1 + b_2)s_0$.
- $\mathsf{EncKey}(x,N) \to (k_1 = \alpha + r_1 b_1, k_2 = r_1(xb_1 + b_2))$.
- Pair. Suppose $x \neq y$. We have $k_1 s_0 - \frac{1}{x-y} k_2 s_0 + \frac{1}{x-y} r_1 c_1 = \alpha s_0$.

Selective Symbolic Property of Construction 9.

- $\mathsf{EncBS}(y)$ outputs $(b_1 : -1,\ b_2 : y,\ s_0 : 1)$. Note that $d_1 = d_2 = 1$.
- $\mathsf{EncR}(x,y)$, where $x = y$, outputs $(r_1 : 1,\ \alpha : 1)$.
- We can verify that
  - $\alpha s_0 : (1)(1) = 1 \neq 0$.
  - $c_1 : (y(-1) + y)(1) = 0$
  - $k_1 : 1 + (1)(-1) = 0$, and $k_2 : (1)(x(-1) + y) = 0$.

Co-selective Symbolic Property of Construction 9.

- $\mathsf{EncBR}(x)$ outputs $(b_1 : -1,\ b_2 : x,\ r_1 : 1,\ \alpha : 1)$. Note that $d_1 = d_2 = 1$.
- $\mathsf{EncS}(x,y)$, where $x = y$, outputs $(s_0 : 1)$.
- We can verify that
  - $\alpha s_0 : (1)(1) = 1 \neq 0$.
  - $k_1 : 1 + (1)(-1) = 0$, and $k_2 : (1)(x(-1) + x) = 0$.
  - $c_1 : (y(-1) + x)(1) = 0$.

C.2 Basic Admissible Schemes

The above mentioned PESs for IBE and NIBE are not admissible. One can apply the $\mathsf{Layer\text{-}Trans}$ transformation of Construction 1 to obtain ones, but these would incur two additional elements for each of the key and ciphertext encoding (counting also the non-lone variables). An alternative way is to use the dual conversion [12,2], which incurs only one element to each encoding. Applying the dual conversion to a *symmetric* predicate will result in the same predicate. Equality (IBE) and inequality (NIBE) predicates are symmetric, hence we can apply without changing their functionalities. We note that the converted PES from the dual conversion can be easily shown to be admissible (by slightly adapting the proof in [2]). For concreteness, we write the dually converted schemes for Construction 8 and 9 here, as we will use them for constructing a PES for KP-NSP in §D.

Construction 10. A PES for P^{IBE}, dually converted from Construction 8.

- $\mathsf{Param} \to 3$. Denote $\mathbf{b} = (b_1, b_2, b_3)$.
- $\mathsf{EncKey}(x, N) \to (k_1 = \alpha + r_1 b_1, k_2 = r_1(y b_2 + b_3))$.
- $\mathsf{EncCt}(y, N) \to c_1 = b_1 s_0 + (x b_2 + b_3) s_1$.
- Pair. Suppose $x = y$. We have $k_1 s_0 - r_1 c_1 + k_2 s_1 = \alpha s_0$.

Construction 11. A PES for P^{NIBE}, dually converted from Construction 9.

- $\mathsf{Param} \to 3$. Denote $\mathbf{b} = (b_1, b_2, b_3)$.
- $\mathsf{EncKey}(x, N) \to (k_1 = \alpha + r_1 b_1, k_2 = r_1(x b_2 + b_3))$.
- $\mathsf{EncCt}(y, N) \to (c_1 = b_1 s_0 + b_2 s_1, c_2 = (y b_2 + b_3) s_1)$.
- Pair. Suppose $x \neq y$. We have $k_1 s_0 - r_1 c_1 + \frac{1}{y-x} r_1 c_2 - \frac{1}{y-x} k_2 s_1 = \alpha s_0$.

D PES for KP-ABE for Non-monotone Span Programs

In this section, for self-containment, we describe a concrete PES construction for $P^{\mathsf{KP\text{-}NSP}}$. We first give the explicit definition of this predicate, which is induced from what we define exactly as $\mathsf{KP1}[P^{\mathsf{TIBBE}}]$ in §9.

Definition 11. The predicate family of *completely unbounded KP-ABE for non-monotone span programs*, $P^{\mathsf{KP\text{-}NSP}} = \{\, P_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \to \{0,1\} \,\}_\kappa$, is indexed by $\kappa = (N)$ and is defined by

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

$$\mathbf{A}|_Y := \left\{\, \mathbf{A}_{i:} \;\middle|\; \big(\pi_1(i) = 1 \wedge \pi_2(i) \in Y\big) \ \vee\ \big(\pi_1(i) = 2 \wedge \pi_2(i) \notin Y\big) \,\right\}.$$

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

We now describe a concrete PES construction for $P^{\mathsf{KP\text{-}NSP}}$, following the explanation in §9. We first recall that $P^{\mathsf{KP\text{-}NSP}} = \mathsf{KP1}[P^{\mathsf{TIBBE}}]$ and that P^{TIBBE} can be embedded into $\mathsf{CP1}_{\mathsf{OR}}[P^{\mathsf{IBBE}} \odot P^{\mathsf{IBR}}]$. Also, we have that P^{IBBE} can be embedded into $\mathsf{CP1}_{\mathsf{OR}}[P^{\mathsf{IBE}}]$, and P^{IBR} can be embedded into $\mathsf{CP1}_{\mathsf{AND}}[P^{\mathsf{NIBE}}]$. We thus obtain a PES for $P^{\mathsf{KP\text{-}NSP}}$ via the following sequence of transformations applying to $\mathit{\Gamma}_{\mathsf{IBE}}$, $\mathit{\Gamma}_{\mathsf{NIBE}}$, which are the PES constructions 10 and 11, respectively.

$$\begin{array}{lll}
\mathit{\Gamma}_{\mathsf{IBBE}} & \leftarrow \mathsf{CP1\text{-}Trans}(\mathit{\Gamma}_{\mathsf{IBE}}). & \text{Confine to OR.} \\
\mathit{\Gamma}_{\mathsf{IBR}} & \leftarrow \mathsf{CP1\text{-}Trans}(\mathit{\Gamma}_{\mathsf{NIBE}}). & \text{Confine to AND.} \\
\mathit{\Gamma}' & \leftarrow \mathsf{Concat\text{-}Trans}(\mathit{\Gamma}_{\mathsf{IBBE}}, \mathit{\Gamma}_{\mathsf{IBR}}). & \\
\mathit{\Gamma}_{\mathsf{TIBBE}} & \leftarrow \mathsf{CP1\text{-}Trans}(\mathit{\Gamma}'). & \text{Confine to OR.} \\
\mathit{\Gamma}_{\mathsf{KP\text{-}NSP}} & \leftarrow \mathsf{KP1\text{-}Trans}(\mathit{\Gamma}_{\mathsf{TIBBE}}). &
\end{array}$$

Construction 12. A PES $\mathit{\Gamma}_{\mathsf{KP\text{-}NSP}}$ for $P^{\mathsf{KP\text{-}NSP}}$.

- $\mathsf{Param} \to 8$. Denote $\mathbf{b} = (b_1, b_2, b_3, \bar{b}_1, \bar{b}_2, \bar{b}_3, g_1, g_2)$.
- $\mathsf{EncCt}(Y = \{y_1, \dots, y_t\}, N) \to \mathbf{c}(\mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) = \big((c_j, \bar{c}_{1,j}, \bar{c}_{2,j})_{j\in[t]}, c', \bar{c}'\big)$:

$$\begin{aligned} c_j &= b_1 s + (y_j b_2 + b_3) s_j, & c' &= g_2 s_0 + g_1 s, \\ \bar{c}_{1,j} &= v_j + \bar{b}_2 \bar{s}_j, & \bar{c}_{2,j} &= (y_j \bar{b}_2 + \bar{b}_3)\bar{s}_j, \quad \bar{c}' = g_2 s_0 + g_1 \bar{s}, \end{aligned}$$

  where $v_1 = \bar{b}_1 \bar{s} - (v_2 + \cdots + v_t)$ and $\mathbf{s} = \big(s_0, s, \bar{s}, (s_j, \bar{s}_j)_{j\in[t]}\big)$, $\hat{\mathbf{s}} = (v_2, \dots, v_t)$. Note that s_0 is special non-lone variable.
- $\mathsf{EncKey}((\mathbf{A}, \pi), N)$. Parse $\mathbf{A} \in \mathbb{Z}_N^{m\times\ell}$ and $\pi : [m] \to \{1, 2\} \times \mathbb{Z}_N$. For $k \in \{1, 2\}$, let $S_k := \{\, i \in [m] \mid \pi_1(i) = k \,\}$. For $i \in [m]$, denote $x_i := \pi_2[i]$. Define

  - for $i \in S_1$: $k_{1,i} = r_i' g_1 + r_i b_1$, $k_{2,i} = r_i(x_i b_2 + b_3)$.
  - for $i \in S_2$: $\bar{k}_{1,i} = r_i' g_1 + r_i \bar{b}_1$, $\bar{k}_{2,i} = r_i(x_i \bar{b}_2 + \bar{b}_3)$.

  Also define $k_i' = \mathbf{A}_{i:}\boldsymbol{\alpha}^\top + r_i' g_2$. Output the key encoding as $\mathbf{k}(\mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) = \big((k_{1,i}, k_{2,i})_{i\in S_1}, (\bar{k}_{1,i}, \bar{k}_{2,i})_{i\in S_2}, (k_i')_{i\in[m]}\big)$, where we let $\mathbf{r} = \big(r_i', r_i\big)_{i\in[m]}$, $\hat{\mathbf{r}} = \boldsymbol{\alpha} := (\alpha, \alpha_2, \dots, \alpha_\ell)$. Note that α is the special lone variable.
- Pair. Suppose $(\mathbf{A}, \pi)$ accepts Y. Let $S_1' := \{\, i \in S_1 \mid \pi_2(i) \in Y \,\}$ and $S_2' := \{\, i \in S_2 \mid \pi_2(i) \notin Y \,\}$. Let $S := S_1 \cup S_2$. We have two cases:
  - Consider $i \in S_1'$. Let $j_i \in [t]$ be such that $y_{j_i} = x_i$. We have

$$r_i' c' - k_{1,i} s + r_i c_{j_i} - k_{2,i} s_{j_i} = r_i' g_2 s_0$$

  - Consider $i \in S_2'$. We have that for all $j \in [t]$, $y_j \neq x_i$. We have

$$r_i' \bar{c}' - \bar{k}_{1,i} \bar{s} + \sum_{j\in[t]} (r_i \bar{c}_{1,j} - \frac{1}{y_j - x_i} r_i \bar{c}_{2,j} + \frac{1}{y_j - x_i} \bar{k}_{2,i} \bar{s}_j) = r_i' g_2 s_0.$$

  Consider a further linear combination for all $i \in S$:

$$k_i' s_0 - (r_i' g_2 s_0) = (\mathbf{A}_{i:}\boldsymbol{\alpha}^\top) s_0.$$

  Now since $\mathbf{1}_1^\ell \in \mathrm{span}(\mathbf{A}|_Y)$, we have linear combination coefficients $\{\, \tau_i \,\}_{i\in S}$ such that $\sum_{i\in S} \tau_i \mathbf{A}_{i:} = \mathbf{1}_1^\ell$. Therefore, the above terms can be linearly combined to $\sum_{i\in S} \tau_i (\mathbf{A}_{i:}\boldsymbol{\alpha}^\top) s_0 = \alpha s_0$, as required.

Theorem 4. *$\Gamma_{\mathsf{KP\text{-}NSP}}$ satisfies* $\mathsf{Sym\text{-}Prop}^+$.

Proof (sketch). This is since the base PESs Γ_{IBE}, Γ_{NIBE} are admissible. We can consecutively apply CP1-Trans and Concat-Trans which preserve admissibility. We can finally apply KP1-Trans since it is admissible. The result of KP1-Trans transformed PES satisfies $\mathsf{Sym\text{-}Prop}^+$. □

On Constant-size Schemes. Recall the notation that w_1, w_2, w_3 represent the number of non-lone variables, lone variables, and polynomials in $\mathsf{ct\text{-}enc}$, and m_1, m_2, m_3 represent the number of non-lone variables, lone variables, and polynomials in $\mathsf{key\text{-}enc}$, respectively. The transformations in [2] can convert a PES $\mathit{\Gamma}$ to another PES $\mathit{\Gamma}'$ with $(w_1', w_2', w_3') = (1, 0, 1)$ (and hence would yield an ABE scheme with constant-size ciphertexts). This comes with a cost that $(n', m_1', m_2', m_3') = ((W_1 n + W_2)W_3, m_1 W_3, m_2 W_1, m_3 W_1 + m_1 W_3^2(W_1 n + W_2))$, where W_1, W_2, W_3 are the bounds for w_1, w_2, w_3 allowed in the original scheme. Schemes with constant-size keys can also be achieved and its resulting efficiency can be obtained swapping the above "m-terms" and "w-terms".

When applying these transformation to our instantiation for KP-ABE for NSP, we have $n = O(1)$, $w_1 = O(t), w_2 = O(t), w_3 = O(t)$, and $m_1 = O(m), m_2 = O(\ell), m_3 = O(m)$, where t is the attribute set size, and $m \times \ell$ is the size of a span program. Bounding t, m, ℓ to T, M, L respectively, we obtain schemes with constant-size ciphertexts or keys as shown in Table 1 and 2.

Revisiting the Okamoto-Takashima Definition. The Okamoto-Takashima (OT) type ABE [33,34] for non-monotone span program was defined differently. We recast it here in our terminology, and explain how to achieve a PES for it modularly.

Define $P^{\mathsf{OT}} : (\{1,2\} \times \mathbb{N} \times \mathcal{U}) \times \mathcal{U}^* \to \{0,1\}$ by $P^{\mathsf{OT}}((i, j, x), (y_1, \ldots, y_t)) = 1 \Leftrightarrow (i = 1 \wedge x = y_j) \vee (i = 2 \wedge x \neq y_j)$. The OT-type ABE for NSP can then be defined as $P^{\mathsf{KP\text{-}NSP\text{-}OT}} := \mathsf{KP1}[P^{\mathsf{OT}}]$.[18] Comparing to normal $P^{\mathsf{KP\text{-}NSP}}$ which takes a *set* of attributes as a ciphertext attribute, this variant takes a *vector* of attributes and requires a key attribute to specify a position j in that vector. Decoupling this basic predicate behind the OT variant of ABE for NSP gives some insight not only how the functionality differs from the original definition but also how to construct a scheme for it. It is not difficult to see that the above P^{OT} can be embedded into $\mathsf{CP1}_{\mathsf{OR}}[P^{\mathsf{IBBE}} \odot \mathsf{CP1}_{\mathsf{OR}}[P^{\mathsf{IBE}} \wedge P^{\mathsf{NIBE}}]]$, where $\wedge$ is the static AND composition (can be obtained by just fixing policy to AND). Intuitively, the two parts of the direct sum are for mode $i = 1$ and $i = 2$, respectively. Inside the second part, the P^{IBE} is used for the equality check on the index j, while P^{NIBE} is used for checking $x \neq y_j$. Interestingly, for the negative attribute part, it uses OR (among $\mathbf{y}$), as opposed to AND (among Y) as in P^{IBR}. OR suffices since the evaluation needs to check only the specified position j.

With the definition in place, we can apply appropriate transformations to achieve a PES for $P^{\mathsf{KP\text{-}NSP\text{-}OT}}$. We omit its explicit description here. The resulting instantiations have the same asymptotic efficiency as shown in our original instantiations in Table 1, 2.

[18] As a caveat, our $\mathsf{KP1}$ transformation is completely unbounded, while the OT12 scheme in [34] requires a bound on the attribute multi-use.

E Proof for Layer Transformation

This section provides the correctness and the proof of Lemma 1 for the symbolic property of the layer transformation, $\mathsf{Layer\text{-}Trans}$, deferred from §4.

Selective Symbolic Property of Construction 1.

- $\mathsf{EncB}'(y)$. Run $\mathsf{EncB}(y) \to (\mathbf{B}_j)_{j\in[n]}$. Note that $\mathbf{B}_j \in \mathbb{Z}_N^{d_1\times d_2}$. Let $d_1' = d_1+1$. Set $\mathbf{B}_j' = \left(\begin{smallmatrix} 0 \\ \mathbf{B}_j \end{smallmatrix}\right) \in \mathbb{Z}_N^{d_1'\times d_2}$, $\mathbf{F}_1 = \mathbf{1}_{1,1}^{d_1'\times d_2}$, and $\mathbf{F}_2 = -\mathbf{1}_{1,1}^{d_1'\times d_2}$.
- $\mathsf{EncS}'(y)$. Set $\mathbf{s}_{\text{new}} = \mathbf{1}_1^{d_2}$. Run $\mathsf{EncS}(y) \to \big((\mathbf{s}_t)_{t\in[w_1]}, (\hat{\mathbf{s}}_z)_{z\in[w_2]}\big)$. Let $q = 1/s_0[1]$. Note that this can be computed since $\mathbf{a}(\mathbf{s}_0)^\top = s_0[1] \neq 0$ due to symbolic property. Set $\mathbf{s}_t' = q\mathbf{s}_t$, $\hat{\mathbf{s}}_z' = q(0, \hat{\mathbf{s}}_z)$. Output $\big(\mathbf{s}_{\text{new}}, (\mathbf{s}_t')_{t\in[w_1]}, (\hat{\mathbf{s}}_z')_{z\in[w_2]}\big)$.
- $\mathsf{EncR}'(x, y)$. Run $\mathsf{EncR}(x, y) \to \big((\mathbf{r}_v)_{v\in[m_1]}, \mathbf{a}, (\hat{\mathbf{r}}_u)_{u\in[m_2]}\big)$. Set $\mathbf{r}_v' = (1, \mathbf{r}_v)$, $\mathbf{r}_{\text{new}} = -\mathbf{1}_1^{d_1'}$, and $\mathbf{a}_{\text{new}} = \mathbf{a} = \mathbf{1}_1^{d_2}$. Output $\big(\mathbf{r}_{\text{new}}, (\mathbf{r}_v')_{v\in[m_1]}, \mathbf{a}_{\text{new}}, (\hat{\mathbf{r}}_u)_{u\in[m_2]}\big)$.

We can verify that $\alpha_{\text{new}} s_{\text{new}} : \mathbf{1}_1^{d_2}(\mathbf{1}_1^{d_2})^\top = 1 \neq 0$ and

- $f_1 s_{\text{new}} + f_2 s_0 : \mathbf{1}_{1,1}^{d_1'\times d_2}(\mathbf{1}_1^{d_2})^\top - \mathbf{1}_{1,1}^{d_1'\times d_2} q(\mathbf{s}_0)^\top = 0$
- $\alpha_{\text{new}} + r_{\text{new}} f_1 : \mathbf{1}_1^{d_2} - \mathbf{1}_1^{d_1'}\mathbf{1}_{1,1}^{d_1'\times d_2} = 0.$
- $\mathbf{c} : 0$ due to the symbolic property of $\mathit{\Gamma}$, and use $\mathbf{B}_j'(\mathbf{s}_t')^\top = q\begin{pmatrix} 0 \\ \mathbf{B}_j(\mathbf{s}_t)^\top \end{pmatrix}$.
- $\tilde{\mathbf{k}} : 0$ due to the symbolic property of $\mathit{\Gamma}$, and use $\mathbf{r}_v'\mathbf{B}_j' = \mathbf{r}_v\mathbf{B}_j$.
- (P6): we have $\mathbf{F}_1 = \mathbf{1}_{1,1}^{d_1'\times d_2}$, $\mathbf{s}_{\text{new}} = \mathbf{1}_1^{d_2}$, and $\mathbf{r}_v'[1] = 1 \neq 0$ for $v \in [m_1]$. (P4) and (P5) are straightforward. (Note that f_1 is treated as b_1 of Def. 4.)

Co-selective Symbolic Property of Construction 1. This is exactly the same as selective argument as above except that now we consider $\mathsf{EncB}'(x)$, $\mathsf{EncR}'(x)$, $\mathsf{EncS}'(x, y)$ which utilizes $\mathsf{EncB}(x)$, $\mathsf{EncR}(x)$, $\mathsf{EncS}(x, y)$, respectively.

F Verifying Proof for Ciphertext-policy Augmentation

In this section, we provide a more in-depth explanation for verifying the selective symbolic property of the $\mathsf{CP1\text{-}Trans}$ conversion, deferred from the proof of Theorem 1 in §5.

Verifying Selective Symbolic Property. First, we can verify that $\mathbf{a}_{\text{new}}\mathbf{s}_{\text{new}}^\top = \mathbf{1}_1^{d_2'}(\mathbf{1}_1^{d_2'})^\top = 1$, which is not zero, as required.

We then verify that each polynomial in $\mathbf{k}$, $\mathbf{c}'$ evaluates to 0. For $\mathsf{ct\text{-}enc}$ $\mathbf{c}'$, let $w_{3,i}$ is the size of $\mathbf{c}^{(i)}$. For $i \in [m]$, $p \in [w_{3,i}]$, the p-th polynomial in $\mathbf{c}'^{(i)}$ is

$$c_p'^{(i)} = \sum_{z\in[w_{2,i}]} \eta_{p,z}^{(i)} \hat{s}_z'^{(i)} + \eta_{p,0,1}^{(i)} \mathbf{A}_{i:}\mathbf{v}^\top + \sum_{t\in[w_{1,i}], j\in[2,n]} \eta_{p,t,j}^{(i)} b_j s_t'^{(i)}. \qquad (20)$$

where we recall that the term $b_1 s_0'^{(i)}$ is replaced by $\mathbf{A}_{i:}\mathbf{v}^\top$ in the construction. First, $\mathbf{A}_{i:}\mathbf{v}^\top$ is substituted and evaluated to

$$\mathbf{A}_{i,1}\mathbf{B}_1'(\mathbf{s}_{\text{new}})^\top + \mathbf{A}_{i,2}\mathbf{v}_2' + \cdots + \mathbf{A}_{i,\ell}\mathbf{v}_\ell' = \sum_{j=1}^{\ell} \mathbf{A}_{i,j}\mathbf{1}_j^{d_1'} = (\mathbf{A}_{i:}, 0, \ldots, 0)^\top \tag{21}$$

Next, $b_j s_t'^{(i)}$ is substituted and evaluated to

$$\mathbf{B}_j'(\mathbf{s}_t'^{(i)})^\top = \left(\begin{array}{cccc} \mathbf{e}_j^{(1)}\mathbf{A}_{1,1} & & \cdots & \mathbf{e}_j^{(m)}\mathbf{A}_{m,1} \\ \vdots & & & \vdots \\ \mathbf{e}_j^{(1)}\mathbf{A}_{1,\ell} & & \cdots & \mathbf{e}_j^{(m)}\mathbf{A}_{m,\ell} \\ \hline \tilde{\mathbf{B}}_j^{(1)} & & & \\ & \tilde{\mathbf{B}}_j^{(2)} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{B}}_j^{(m)} \end{array}\right) \begin{pmatrix} 0 \\ \vdots \\ 0 \\ (\mathbf{s}_t^{(i)})^\top \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left(\begin{array}{c} (\mathbf{e}_j^{(i)}(\mathbf{s}_t^{(i)})^\top)\mathbf{A}_{i,1} \\ \vdots \\ (\mathbf{e}_j^{(i)}(\mathbf{s}_t^{(i)})^\top)\mathbf{A}_{i,\ell} \\ \hline 0 \\ \vdots \\ 0 \\ \tilde{\mathbf{B}}_j^{(i)}(\mathbf{s}_t^{(i)})^\top \\ 0 \\ \vdots \\ 0 \end{array}\right). \tag{22}$$

where the appearances of $(\mathbf{s}_t^{(i)})^\top$ and $\tilde{\mathbf{B}}_j^{(i)}(\mathbf{s}_t^{(i)})^\top$ are in their respective i-th block. From these, together with the substitution for $\hat{s}_z'^{(i)}$ to $(\hat{\mathbf{s}}_z'^{(i)})^\top$ via Eq. (14), we have that $c_p'^{(i)}$ is substituted and evaluated to

$$\sum_{z\in[w_{2,i}]} \eta_{p,z}^{(i)} \left(\begin{array}{c} (\hat{\mathbf{s}}_z^{(i)}[1])^\top \mathbf{A}_{i,\ell} \\ \vdots \\ (\hat{\mathbf{s}}_z^{(i)}[1])^\top \mathbf{A}_{i,\ell} \\ \hline 0 \\ \vdots \\ 0 \\ (\hat{\mathbf{s}}_z^{(i)}[2,d_1])^\top 0 \\ \vdots \\ 0 \end{array}\right) + \eta_{p,0,1}^{(i)} \left(\begin{array}{c} \mathbf{A}_{i,1} \\ \vdots \\ \mathbf{A}_{i,\ell} \\ \hline 0 \\ \vdots \\ 0 \end{array}\right) + \sum_{\substack{t\in[w_{1,i}] \\ j\in[2,n]}} \eta_{p,t,j}^{(i)} \left(\begin{array}{c} (\mathbf{e}_j^{(i)}(\mathbf{s}_t^{(i)})^\top)\mathbf{A}_{i,1} \\ \vdots \\ (\mathbf{e}_j^{(i)}(\mathbf{s}_t^{(i)})^\top)\mathbf{A}_{i,\ell} \\ \hline 0 \\ \vdots \\ 0 \\ \tilde{\mathbf{B}}_j^{(i)}(\mathbf{s}_t^{(i)})^\top \\ 0 \\ \vdots \\ 0 \end{array}\right), \tag{23}$$

which is a vector in $\mathbb{Z}_N^{d_1'\times 1}$. We argue that each row of it is 0. To see this, we use the symbolic property of Γ, which ensures that the substitution of $c_p^{(i)}$ via

$\mathsf{EncBS}(\pi(i))$, as shown below, will evaluate to 0:

$$\sum_{z\in[w_{2,i}]} \eta_{p,z}^{(i)}(\hat{\mathbf{s}}_z^{(i)})^\top + \eta_{p,0,1}^{(i)}\mathbf{B}_1\mathbf{s}_0^{(i)} + \sum_{t\in[w_{1,i}],j\in[2,n]} \eta_{p,t,j}^{(i)}\mathbf{B}_j(\mathbf{s}_t^{(i)})^\top \tag{24}$$

$$= \sum_{z\in[w_{2,i}]} \eta_{p,z}^{(i)} \begin{pmatrix} \hat{\mathbf{s}}_z^{(i)}[1] \\ (\hat{\mathbf{s}}_z^{(i)}[2,d_1])^\top \end{pmatrix} + \eta_{p,0,1}^{(i)}(\mathbf{1}_1^{d_1})^\top + \sum_{\substack{t\in[w_{1,i}]\\ j\in[2,n]}} \eta_{p,t,j}^{(i)} \begin{pmatrix} \mathbf{e}_j^{(i)}(\mathbf{s}_t^{(i)})^\top \\ \tilde{\mathbf{B}}_j^{(i)}(\mathbf{s}_t^{(i)})^\top \end{pmatrix}. \tag{25}$$

One can observe that the element in position j, for $j \in [1,\ell]$, in Eq. (23) is exactly the element in the first row of Eq. (25) multiplied by $\mathbf{A}_{i,j}$. The i-th block of Eq. (23) is exactly the remaining part of Eq. (25) when excluding the first row.

For $\mathsf{key\text{-}enc}$ $\mathbf{k}$, let m_3 is the size of $\mathbf{k}$. The first polynomial $k_1 = \alpha + r_1 b_1$ is substituted to $\mathbf{a}_{\text{new}} + \mathbf{r}'_1\mathbf{B}'_1$, which is evaluated to 0 since $\mathbf{a}_{\text{new}} = \mathbf{1}_1^{d'_2}$, $\mathbf{B}'_1 = \mathbf{1}_{1,1}^{d'_1\times d'_2}$ and the first element of $\mathbf{r}'_1$ is $-\omega_1 = -1$. Next, we consider the p-th polynomial in $\mathbf{k}$, for $p \in [2, m_3]$, which is

$$k_p = \sum_{u\in[m_2]} \phi_{p,u}\hat{r}_u + \sum_{v\in[m_1],j\in[2,n]} \phi_{p,v,j} r_v b_j. \tag{26}$$

Note that α, b_1 do not appear due to admissibility. We then observe that $r_v b_j$ is substituted and evaluated to

$$\mathbf{r}'_v\mathbf{B}'_j = -\Big(\boldsymbol{\omega},\ g_1\mathbf{r}_v^{(1)}[2,d_1], \ldots, g_m\mathbf{r}_v^{(m)}[2,d_1]\Big) \left(\begin{array}{cccc} \mathbf{e}_j^{(1)}\mathbf{A}_{1,1} & & \cdots & \mathbf{e}_j^{(m)}\mathbf{A}_{m,1} \\ \vdots & & & \vdots \\ \mathbf{e}_j^{(1)}\mathbf{A}_{1,\ell} & & \cdots & \mathbf{e}_j^{(m)}\mathbf{A}_{m,\ell} \\ \hline \tilde{\mathbf{B}}_j^{(1)} & & & \\ & \tilde{\mathbf{B}}_j^{(2)} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{B}}_j^{(m)} \end{array}\right)$$

$$= -\Big(\mathbf{e}_j^{(1)}\mathbf{A}_{1:}\boldsymbol{\omega}^\top + g_1\mathbf{r}_v^{(1)}[2,d_1]\tilde{\mathbf{B}}_j^{(1)}, \ \ldots\ , \mathbf{e}_j^{(m)}\mathbf{A}_{m:}\boldsymbol{\omega}^\top + g_m\mathbf{r}_v^{(m)}[2,d_1]\tilde{\mathbf{B}}_j^{(m)}\Big)$$

$$= -\Big(g_1\mathbf{r}_v^{(1)}\mathbf{B}_j^{(1)}, \ \ldots\ , g_m\mathbf{r}_v^{(m)}\mathbf{B}_j^{(m)}\Big) \tag{27}$$

This, together with our substitution of $\hat{r}_u$ to $\hat{\mathbf{r}}'_u$ in Eq. (16), we have that the substitution of k_p is a vector in $\mathbb{Z}_N^{1\times d'_2}$ of which the i-th block of length d_2, for $i \in [m]$, is exactly

$$g_i(\mathbf{u}_i) \coloneqq g_i\Big(\sum_{u\in[m_2]} \phi_{p,u}\hat{\mathbf{r}}_u^{(i)} + \sum_{v\in[m_1],j\in[2,n]} \phi_{p,v,j}\mathbf{r}_v^{(i)}\mathbf{B}_j^{(i)}\Big).$$

For all $i \in [m]$, this i-th block is evaluated to exactly 0 since

- if $i \in S$, then we have $g_i = 0$,
- if $i \notin S$, then the polynomials in the i-th block, $g_i(\mathbf{u}_i)$, evaluate to 0. This is since $\mathbf{u}_i$ is exactly the substitution result for k_p via $\mathsf{EncR}(x, \pi(i))$, and the selective symbolic property of $\mathit{\Gamma}$ applies here since, in this case of i, we have $P_\kappa(x, \pi(i)) = 0$.

G Definition and Proof for Key-policy Augmentation

G.1 Definition

In §6, we define $\mathsf{KP1}[P]$ based on the duality notion (§3.1) in a back-and-forth manner. For concreteness and self-containment, we also describe its explicit definition here.

Definition 12. 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-span-program-augmented predicate* over P as $\mathsf{KP1}[P] = \left\{\, \bar{P}_\kappa \,\right\}_\kappa$ where $\bar{P}_\kappa : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{\, 0, 1 \,\}$ by letting

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

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

G.2 Proof for Key-policy Augmentation

We describe the omitted proof of Theorem 2, deferred from §6.

Co-selective Symbolic Property. We first prove co-selective symbolic property of $\mathit{\Gamma}'$ from that of $\mathit{\Gamma}$. We assume w.l.o.g. that $m \geq \ell$ (see Remark 5). We define the following algorithms.

- $\mathsf{EncBR}'(\mathbf{A}, \pi)$. For each $i \in [m]$, run

$$\mathsf{EncBR}(\pi(i)) \to \left(\mathbf{B}_1^{(i)}, \dots, \mathbf{B}_n^{(i)};\ \mathbf{r}_1^{(i)}, \dots, \mathbf{r}_{m_{1,i}}^{(i)};\ \mathbf{a}, \hat{\mathbf{r}}_1^{(i)}, \dots, \hat{\mathbf{r}}_{m_{2,i}}^{(i)}\right),$$

  where $\mathbf{B}_j^{(i)} \in \mathbb{Z}_N^{d_1 \times d_2}$, $\mathbf{r}_v^{(i)} \in \mathbb{Z}_N^{1 \times d_1}$, $\hat{\mathbf{r}}_u^{(i)} \in \mathbb{Z}_N^{1 \times d_2}$, and $\mathbf{a} = \mathbf{1}_1^{d_2} \in \mathbb{Z}_N^{1 \times d_2}$. Also, from admissibility we have $\mathbf{B}_1^{(i)} = \mathbf{B}_1 = \mathbf{1}_{1,1}^{d_1 \times d_2}$. Let $d_1' = md_1$ and $d_2' = md_2$. Any vector of length d_1' can be naturally divided into m blocks, each with length d_1. (The same goes for d_2'.) We then let

$$\begin{aligned}
\mathbf{B}_1' &= \left(\begin{array}{ccc|c} \mathbf{A}_{1,1}\mathbf{B}_1 & \cdots & \mathbf{A}_{1,\ell}\mathbf{B}_1 & \\ \vdots & & \vdots & 0 \\ \mathbf{A}_{m,1}\mathbf{B}_1 & \cdots & \mathbf{A}_{m,\ell}\mathbf{B}_1 & \end{array}\right) && \in \mathbb{Z}_N^{d_1' \times d_2'}, && (28)\\
\mathbf{B}_j' &= \begin{pmatrix} \mathbf{B}_j^{(1)} & & & \\ & \mathbf{B}_j^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{B}_j^{(m)} \end{pmatrix} && \in \mathbb{Z}_N^{d_1' \times d_2'}, && (29)\\
\mathbf{r}_v'^{(i)} &= (0, \ldots, 0, \overset{\overset{\text{block } i}{\downarrow}}{\mathbf{r}_v^{(i)}}, 0, \ldots, 0) && \in \mathbb{Z}_N^{1 \times d_1'}, &&\\
\mathbf{a}_{\text{new}} &= \mathbf{1}_1^{d_2'} && \in \mathbb{Z}_N^{1 \times d_2'}, &&\\
\mathbf{v}_\iota' &= (0, \ldots, 0, \overset{\overset{\text{block } \iota}{\downarrow}}{\mathbf{1}_1^{d_2}}, 0, \ldots, 0) = \mathbf{1}_{(\iota-1)d_2+1}^{d_2'} && \in \mathbb{Z}_N^{1 \times d_2'}, &&\\
\hat{\mathbf{r}}_u'^{(i)} &= (0, \ldots, 0, \overset{\overset{\text{block } i}{\downarrow}}{\hat{\mathbf{r}}_u^{(i)}}, 0, \ldots, 0) && \in \mathbb{Z}_N^{1 \times d_2'}, && (30)
\end{aligned}$$

for $j \in [2, n]$, $i \in [m]$, $v \in [m_{1,i}]$, $\iota \in [2, \ell]$, $u \in [m_{2,i}]$. The block number i indicates that the specified sub-vector is at the i-th block position of the whole vector.[19] Finally, EncBR' outputs

$$\left(\mathbf{B}_1', \ldots, \mathbf{B}_n'; \left(\mathbf{r}_1'^{(i)}, \ldots, \mathbf{r}_{m_{1,i}}'^{(i)}\right)_{i \in [m]}; \mathbf{a}_{\text{new}}, \mathbf{v}_2', \ldots, \mathbf{v}_\ell', \left(\hat{\mathbf{r}}_1'^{(i)}, \ldots, \hat{\mathbf{r}}_{m_{2,i}}'^{(i)}\right)_{i \in [m]}\right).$$

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

1. For each $i \notin S$, we have that $P_\kappa(\pi(i), y) = 0$. Hence, it is possible to run

$$\mathsf{EncS}(\pi(i), y) \to \left(\mathbf{s}_0^{(i)}, \ldots, \mathbf{s}_{w_1}^{(i)};\ \hat{\mathbf{s}}_1^{(i)}, \ldots, \hat{\mathbf{s}}_{w_2}^{(i)}\right),$$

   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. Since $\bar{P}_\kappa((\mathbf{A}, \pi), y) = 0$, 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 $i \in S$. We denote $q_i = \mathbf{A}_{i:}\boldsymbol{\omega}^\top$ for all $i \in [m]$.
3. Let $\mathbf{s}_0' = (\omega_1 \mathbf{1}_1^{d_2}, \ldots, \omega_\ell \mathbf{1}_1^{d_2}, 0, \ldots, 0) \in \mathbb{Z}_N^{1 \times d_2'}$. For $t \in [w_1]$, $z \in [w_2]$, let

$$\begin{aligned}
\mathbf{s}_t' &= \left(q_1 \mathbf{s}_t^{(1)}, \ldots, q_m \mathbf{s}_t^{(m)}\right) \in \mathbb{Z}_N^{1 \times d_2'}, \\
\hat{\mathbf{s}}_z' &= \left(q_1 \hat{\mathbf{s}}_z^{(1)}, \ldots, q_m \hat{\mathbf{s}}_z^{(m)}\right) \in \mathbb{Z}_N^{1 \times d_1'}. \qquad (31)
\end{aligned}$$

   Finally, output $\left(\mathbf{s}_0', \mathbf{s}_1', \ldots, \mathbf{s}_{w_1}';\ \hat{\mathbf{s}}_1', \ldots, \hat{\mathbf{s}}_{w_2}'\right)$.

[19] For a d_1'-length vector, the i-th block consists of position $(i-1)d_1 + 1$ to id_1. For a d_2'-length vector, the i-th block consists of position $(i-1)d_2 + 1$ to id_2.

Remark 5. In the above, we assume w.l.o.g. that $m \geq \ell$. In the case $m < \ell$, we simply let $d_2' = \ell d_2$ and append 0 to the right of $\mathbf{B}_j'$ (for $j \in [2, n]$) instead of $\mathbf{B}_1'$.

Verifying Properties. First, we can verify that, since $\omega_1 = 1$, we have that $\mathbf{a}_{\text{new}}(\mathbf{s}_0')^\top = 1$, which is not zero, as required.

We then verify that each polynomial in $\mathbf{k}'$, $\mathbf{c}$ evaluates to 0. For $\mathsf{key\text{-}enc}$ $\mathbf{k}'$, let $m_{3,i}$ is the size of $\mathbf{k}^{(i)}$. For $i \in [m]$, $p \in [2, m_{3,i}]$, the p-th polynomial in $\mathbf{k}'^{(i)}$ is

$$\sum_{u \in [m_{2,i}]} \phi_{p,u}^{(i)} \hat{r}_u^{(i)} + \sum_{v \in [m_{1,i}], j \in [2,n]} \phi_{p,v,j}^{(i)} r_v^{(i)} b_j \tag{32}$$

where we recall that the coefficients are those of $\mathbf{k}^{(i)}$ obtained from $\mathsf{EncKey}(\pi(i), N)$ (for the PES $\mathit{\Gamma}$). Note that α, b_1 do not appear due to admissibility. We then observe that $r_v^{(i)} b_j$ is substituted and evaluated to

$$\mathbf{r}_v'^{(i)} \mathbf{B}_j' = (0, \ldots, 0, \overset{\text{block } i}{\overset{\downarrow}{\mathbf{r}_v^{(i)} \mathbf{B}_j^{(i)}}}, 0, \ldots, 0).$$

This, together with Eq. (30), we have that the substitution for the term (32) have only elements in the i-th block remained, which is

$$\sum_{u \in [m_{2,i}]} \phi_{p,u}^{(i)} \hat{\mathbf{r}}_u^{(i)} + \sum_{v \in [m_{1,i}], j \in [2,n]} \phi_{p,v,j}^{(i)} \mathbf{r}_v^{(i)} \mathbf{B}_j^{(i)}$$

but this is exactly 0 due to the co-selective symbolic property of $\mathit{\Gamma}$.

The remaining elements in $\mathsf{key\text{-}enc}$ are the first polynomials in each $\mathbf{k}'^{(i)}$, namely, $k_1'^{(i)} = \mathbf{A}_{i:} \mathbf{v}^\top + r_1^{(i)} b_1$. First, $\mathbf{A}_{i:} \mathbf{v}^\top$ is substituted and evaluated to

$$\begin{aligned} \mathbf{A}_{i,1} \mathbf{a}_{\text{new}} + \mathbf{A}_{i,2} \mathbf{v}_2' + \cdots + \mathbf{A}_{i,\ell} \mathbf{v}_\ell' &= \sum_{j=1}^{\ell} \mathbf{A}_{i,j} \mathbf{1}_{(j-1)d_2+1}^{d_2'} \\ &= (\mathbf{A}_{i,1} \mathbf{1}_1^{d_2}, \mathbf{A}_{i,2} \mathbf{1}_1^{d_2}, \ldots, \mathbf{A}_{i,\ell} \mathbf{1}_1^{d_2}, 0, \ldots, 0). \end{aligned}$$

The other element, $r_1^{(i)} b_1$, is substituted and evaluated to

$$\begin{aligned} \mathbf{r}_1'^{(i)} \mathbf{B}_1' &= (0, \ldots, 0, \overset{\text{block } i}{\overset{\downarrow}{\mathbf{r}_1^{(i)}}}, 0, \ldots, 0) \left(\begin{array}{ccc|c} \mathbf{A}_{1,1} \mathbf{B}_1 & \cdots & \mathbf{A}_{1,\ell} \mathbf{B}_1 & \\ \vdots & & \vdots & 0 \\ \mathbf{A}_{m,1} \mathbf{B}_1 & \cdots & \mathbf{A}_{m,\ell} \mathbf{B}_1 & \end{array} \right) \\ &= -(\mathbf{A}_{i,1} \mathbf{1}_1^{d_2}, \mathbf{A}_{i,2} \mathbf{1}_1^{d_2}, \ldots, \mathbf{A}_{i,\ell} \mathbf{1}_1^{d_2}, 0, \ldots, 0), \end{aligned}$$

where we use $\mathbf{r}_1^{(i)} \mathbf{B}_1 = -\mathbf{a} = -\mathbf{1}_1^{d_2}$. Hence, the whole substitution for $k_1'^{(i)}$ is 0.

For $\mathsf{ct\text{-}enc}$ $\mathbf{c}$, the p-th polynomial in $\mathbf{c}$ is

$$c_p = \sum_{z \in [w_2]} \eta_{p,z} \hat{s}_z + \eta_{p,0,1} b_1 s_0 + \sum_{t \in [w_1], j \in [2,n]} \eta_{p,t,j} b_j s_t,$$

where we emphasize that b_1, s_0 does not appear except in monomial $b_1 s_0$, due to admissibility.

Via EncS' (and EncBR'), this is substituted to

$$\sum_{z\in[w_2]} \eta_{p,z}(\hat{\mathbf{s}}'_z)^\top + \eta_{p,0,1}\mathbf{B}'_1(\mathbf{s}'_0)^\top + \sum_{t\in[w_1],j\in[2,n]} \eta_{p,t,j}\mathbf{B}'_j(\mathbf{s}'_t)^\top. \tag{33}$$

We examine each term. First, we have

$$\mathbf{B}'_1(\mathbf{s}'_0)^\top = \left(\begin{array}{ccc|c} \mathbf{A}_{1,1}\mathbf{B}_1 & \cdots & \mathbf{A}_{1,\ell}\mathbf{B}_1 & \\ \vdots & & \vdots & 0 \\ \mathbf{A}_{m,1}\mathbf{B}_1 & \cdots & \mathbf{A}_{m,\ell}\mathbf{B}_1 & \end{array}\right) \begin{pmatrix} \omega_1(\mathbf{1}_1^{d_2})^\top \\ \vdots \\ \omega_\ell(\mathbf{1}_1^{d_2})^\top \\ 0 \end{pmatrix} = \begin{pmatrix} q_1(\mathbf{1}_1^{d_1})^\top \\ \vdots \\ q_m(\mathbf{1}_1^{d_1})^\top \end{pmatrix} \tag{34}$$

and for $j \in [2,n], t \in [w_1]$ we have

$$\mathbf{B}'_j(\mathbf{s}'_t)^\top = \begin{pmatrix} \mathbf{B}_j^{(1)} & & \\ & \ddots & \\ & & \mathbf{B}_j^{(m)} \end{pmatrix} \begin{pmatrix} q_1(\mathbf{s}_t^{(1)})^\top \\ \vdots \\ q_m(\mathbf{s}_t^{(m)})^\top \end{pmatrix} = \begin{pmatrix} q_1\mathbf{B}_j^{(1)}(\mathbf{s}_t^{(1)})^\top \\ \vdots \\ q_m\mathbf{B}_j^{(m)}(\mathbf{s}_t^{(m)})^\top \end{pmatrix}.$$

Hence, we have that the term (33) evaluates to

$$\begin{aligned} \begin{pmatrix} q_1(\mathbf{u}_1)^\top \\ \vdots \\ q_m(\mathbf{u}_m)^\top \end{pmatrix} \coloneqq \sum_{z\in[w_2]} \eta_{p,z} \begin{pmatrix} q_1(\hat{\mathbf{s}}_z^{(1)})^\top \\ \vdots \\ q_m(\hat{\mathbf{s}}_z^{(m)})^\top \end{pmatrix} + \begin{pmatrix} q_1(\mathbf{1}_1^{d_1})^\top \\ \vdots \\ q_m(\mathbf{1}_1^{d_1})^\top \end{pmatrix} \\ + \sum_{\substack{t\in[w_1]\\ j\in[2,n]}} \eta_{p,t,j} \begin{pmatrix} q_1\mathbf{B}_j^{(1)}(\mathbf{s}_t^{(1)})^\top \\ \vdots \\ q_m\mathbf{B}_j^{(m)}(\mathbf{s}_t^{(m)})^\top \end{pmatrix}. \end{aligned} \tag{35}$$

Eq. (35) is a vector in $\mathbb{Z}_N^{d'_1\times 1}$. This can be divided to m blocks each of length d_1. For all $i \in [m]$, the i-th block is evaluated to exactly 0 since

- if $i \in S$, then we have $q_i = 0$,
- if $i \notin S$, then the polynomials in the i-th block, $q_i(\mathbf{u}_i)^\top$, evaluate to 0. This is since $(\mathbf{u}_i)^\top$ is exactly the substitution result for c_p via $\mathsf{EncS}(\pi(i), y)$, and the co-selective symbolic property of $\mathit{\Gamma}$ applies here since, in this case of i, we have $P_\kappa(\pi(i), y) = 0$.

Selective Symbolic Property. We prove selective symbolic property of $\mathit{\Gamma}'$ from that of $\mathit{\Gamma}$. We define the following algorithms.

- $\mathsf{EncBS}'(y) = \mathsf{EncBS}(y)$.

- $\mathsf{EncR}'((\mathbf{A},\pi),y)$. First note that we have the condition $\bar{P}_\kappa((\mathbf{A},\pi),y)=0$. Let $S=\{\, i\in[m] \mid P_\kappa(\pi(i),y)=1 \,\}$.
  1. From the condition $\bar{P}_\kappa((\mathbf{A},\pi),y)=0$ and from Proposition 1, we can obtain a vector $\boldsymbol{\omega}=(\omega_1,\dots,\omega_\ell)\in\mathbb{Z}_N^{1\times\ell}$ such that $\omega_1=1$ and $\mathbf{A}_{i:}\boldsymbol{\omega}^\top=0$ for all $i\in S$. Denote $q_i=\mathbf{A}_{i:}\boldsymbol{\omega}^\top$.
  2. For each $i\notin S$, we have that $P_\kappa(\pi(i),y)=0$. Hence, it is possible to run
  $$\mathsf{EncR}(\pi(i),y)\to\left(\mathbf{r}_1^{(i)},\dots,\mathbf{r}_{m_{1,i}}^{(i)};\ \mathbf{a},\hat{\mathbf{r}}_1^{(i)},\dots,\hat{\mathbf{r}}_{m_{2,i}}^{(i)}\right).$$
  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 $\mathbf{a}=\mathbf{1}_1^{d_2}\in\mathbb{Z}_N^{1\times d_2}$.
  3. For $i\in[m]$, $v\in[m_{1,i}]$, $\iota\in[2,\ell]$, $u\in[m_{2,i}]$, let
  $$\begin{aligned}\mathbf{r}_v'^{(i)} &= q_i\,\mathbf{r}_v^{(i)} && \in\mathbb{Z}_N^{1\times d_1'},\\ \mathbf{a}_{\mathrm{new}} &= \omega_1\mathbf{1}_1^{d_2'}=(1,\ 0,\dots,0) && \in\mathbb{Z}_N^{1\times d_2'},\\ \mathbf{v}_\iota' &= \omega_\iota\mathbf{1}_1^{d_2'}=(\omega_\iota,0,\dots,0) && \in\mathbb{Z}_N^{1\times d_2'},\\ \hat{\mathbf{r}}_u'^{(i)} &= q_i\,\hat{\mathbf{r}}_u^{(i)} && \in\mathbb{Z}_N^{1\times d_2'}.\end{aligned}$$
  4. Output $\left(\left(\mathbf{r}_1'^{(i)},\dots,\mathbf{r}_{m_{1,i}}'^{(i)}\right)_{i\in[m]};\mathbf{a}_{\mathrm{new}},\mathbf{v}_2',\dots,\mathbf{v}_\ell',\left(\hat{\mathbf{r}}_1'^{(i)},\dots,\hat{\mathbf{r}}_{m_{2,i}}'^{(i)}\right)_{i\in[m]}\right)$.

Verifying Properties. First we can verify that $\mathbf{a}_{\mathrm{new}}\mathbf{s}_0^\top=\mathbf{1}_1^{d_2}(\mathbf{1}_1^{d_2})^\top=1\neq 0$, as required. Next, since we define $\mathsf{EncBS}'(y)=\mathsf{EncBS}(y)$, the substitution for $\mathsf{ct\text{-}enc}$ is trivially evaluated to 0, due to the selective symbolic property of $\mathit{\Gamma}$.

It remains to consider the substitution for $\mathsf{key\text{-}enc}$ $\mathbf{k}'$. For $i\in[m]$, consider the p-th polynomial $k_p^{(i)}$ where $p\in[2,m_{3,i}]$. Let $\mathbf{u}_i\in\mathbb{Z}_N^{1\times d_2}$ denote the substitution result for $k_p^{(i)}$ (as a part of $\mathbf{k}^{(i)}$) via $\mathsf{EncR}(\pi(i),y)$ (and $\mathsf{EncBS}(y)$). By our constructions of $\mathbf{r}_v'^{(i)}$ and $\hat{\mathbf{r}}_u'^{(i)}$, it is straightforward to see that the substitution for $k_p'^{(i)}$ (as a part of $\mathbf{k}'^{(i)}$) via $\mathsf{EncR}'((\mathbf{A},\pi),y)$ (and $\mathsf{EncBS}'(y)$) is indeed $q_i\mathbf{u}_i$. We can see that $q_i\mathbf{u}_i=0$ since if $i\in S$ then $q_i=0$, while if $i\notin S$, we have $\mathbf{u}_i=0$ due to the selective symbolic property of $\mathit{\Gamma}$.

Finally, the remaining terms consist of $k_1'^{(i)}=\mathbf{A}_{i:}\mathbf{v}^\top+r_1^{(i)}b_1$, for $i\in[m]$. This is substituted and evaluated to 0 as

$$\mathbf{A}_{i,1}\mathbf{a}_{\mathrm{new}}+\mathbf{A}_{i,2}\mathbf{v}_2'+\cdots+\mathbf{A}_{i,\ell}\mathbf{v}_\ell'+\mathbf{r}_1'^{(i)}\mathbf{B}_1'=q_i\mathbf{1}_1^{d_2}+q_i\mathbf{r}_1^{(i)}\mathbf{B}_1=0.$$

H Proofs for Direct Sum

In this section, we provide the proofs omitted from §7. We start with the following lemma.

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

Proof. We first explicitly describe the deduced definition of predicate family $\mathsf{CP1}_{\mathsf{OR}}[\mathsf{DS}[\mathcal{P}]]$: it is specified by $\bar{P}_\kappa : \mathbb{X}_\kappa \times 2^{\mathbb{Y}_\kappa} \to \{\,0, 1\,\}$ where

$$\bar{P}_\kappa\big((i, x),\, Y\big) = 1 \iff \exists (i, y) \in Y \text{ s.t. } P^{(i)}_{\kappa_i}(x, y) = 1.$$

Now, from the definition of $\mathsf{KP1}$ over one predicate family $\mathsf{CP1}_{\mathsf{OR}}[\mathsf{DS}[\mathcal{P}]]$ (using Definition 12), we have that $\mathsf{KP1}[\mathsf{CP1}_{\mathsf{OR}}[\mathsf{DS}[\mathcal{P}]]]$ renders to exactly the same as the definition of $\mathsf{KP}[\mathcal{P}]$ (Definition 6). (Put in other words, the embedding functions from the latter to the former can defined as the identity functions.) □

H.1 Proof for Parameter Concatenation Scheme

Lemma 3 (restated)**.** Suppose that, for all $j \in [k]$, the PES $\Gamma^{(j)}$ 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)-$\mathsf{Sym\text{-}Prop}^{+}$.

Proof (sketch). We prove selective symbolic property by defining substitution algorithms as follows.

- $\mathsf{EncB}'(j, y)$.
  - Run $\mathsf{EncB}^{(j)}(y)$ to obtain $\mathbf{B}^{(j)}_1, \ldots, \mathbf{B}^{(j)}_{n_j}$.
  - For $j' \in [k] \setminus \{j\}$, run $\mathsf{EncB}^{(j')}(\bot)$ to obtain $\mathbf{B}^{(j')}_1, \ldots, \mathbf{B}^{(j')}_{n_{j'}}$.
- $\mathsf{EncS}'(j, y)$. Simply run $\mathsf{EncS}^{(j)}(y)$.
- $\mathsf{EncR}'((i, x), (j, y))$. Since $\bar{P}_\kappa\big((i, x),\, (j, y)\big) = 0$, we have two cases:
  - Case $i = j$ and $P^{(j)}_{\kappa_j}(x, y) = 0$. Output $\mathsf{EncR}^{(j)}(x, y)$.
  - Case $i \neq j$. Output $\mathsf{EncR}^{(i)}(x, \bot)$, which is possible to run for all $x \in \mathcal{X}^{(i)}_\kappa$.

It is then straightforward to see that the two properties follow from those of $\mathsf{Sym\text{-}Prop}^{++}$ of $\Gamma^{(j)}$ for all $j \in [k]$.

Co-selective symbolic property can be proved as follows.

- $\mathsf{EncB}'(i, x)$.
  - Run $\mathsf{EncB}^{(i)}(x)$ to obtain $\mathbf{B}^{(i)}_1, \ldots, \mathbf{B}^{(i)}_{n_i}$.
  - For $i' \in [k] \setminus \{i\}$, simply set $\mathbf{B}^{(i')}_1, \ldots, \mathbf{B}^{(i')}_{n_{i'}}$ to 0.
- $\mathsf{EncR}'(i, x)$. Simply run $\mathsf{EncR}^{(i)}(x)$.
- $\mathsf{EncS}'((i, x), (j, y))$. Since $\bar{P}_\kappa\big((i, x),\, (j, y)\big) = 0$, we have two cases:
  - Case $j = i$ and $P^{(i)}_{\kappa_i}(x, y) = 0$. Output $\mathsf{EncS}^{(i)}(x, y)$.
  - Case $j \neq i$. Output zero vectors for all elements except $\mathbf{s}_0$, which is set so that $\mathbf{a}\mathbf{s}_0^\top \neq 0$.

The property (P1) holds due to our setting of $\mathbf{s}_0^\top$. The property (P2) also holds from the $\mathsf{Sym\text{-}Prop}$ of $\Gamma^{(i)}$ in $\mathsf{key\text{-}enc}$ and the above former case of $\mathsf{ct\text{-}enc}$. For the latter case of $\mathsf{ct\text{-}enc}$, the only non-zero vector $\mathbf{s}_0$ will appear only as a product $\mathbf{B}^{(j)}_\iota \mathbf{s}_0^\top$ which is 0, since $\mathbf{B}^{(j)}_\iota$ for $j \neq i$ was set to 0. □

Lemma 6. *Suppose that Γ for P satisfies (d_1, d_2)-$\mathsf{Sym\text{-}Prop}$. Then,* $\mathsf{Plus\text{-}Trans}(\Gamma)$ *for P satisfies (d_1, d_2)-$\mathsf{Sym\text{-}Prop}^{++}$.*

Proof. $\mathsf{Sym\text{-}Prop}^{+}$ follows from Proposition 3. We prove property (P7) as follows.

- $\mathsf{EncB}(\perp)$. Set all $\mathbf{B}_j$ to 0. Set $\mathbf{F} = \mathbf{1}_{1,1}^{d_1\times d_2} \in \mathbb{Z}_N^{d_1\times d_2}$. Output $(\mathbf{B}_1, \ldots, \mathbf{B}_n, \mathbf{F})$.
- $\mathsf{EncR}(x, \perp)$. Let $\mathbf{r}_{\mathrm{new}} = -\mathbf{1}_1^{d_1}$, and all the remaining vectors be 0.

In $\mathsf{key\text{-}enc}$, $\alpha + r_{\mathrm{new}} f$ is substituted and evaluated to $\mathbf{a} + \mathbf{r}_{\mathrm{new}}\mathbf{F} = \mathbf{1}_1^{d_2} - \mathbf{1}_1^{d_1}\mathbf{1}_{1,1}^{d_1\times d_2} = 0$. As the other remaining terms are all 0, this concludes the proof. □

H.2 Proof for Parameter Reuse Scheme

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

Proof. We first prove selective symbolic property by defining substitution algorithms as follows.

- $\mathsf{EncB}'(j, y)$. Run $\mathsf{EncB}^{(j)}(y)$ to obtain $\mathbf{B}_1, \ldots, \mathbf{B}_{n_j}$. Set $\mathbf{G}_j = -\mathbf{1}_{1,1}^{d_1\times d_2}$. Set $\mathbf{G}_1, \ldots, \mathbf{G}_{j-1}, \mathbf{G}_{j+1}, \ldots, \mathbf{G}_k$ to 0. Set $\mathbf{H}_1, \ldots, \mathbf{H}_k$ to $\mathbf{1}_{1,1}^{d_1\times d_2}$. The remaining, $\mathbf{B}_{n_j+1}, \ldots, \mathbf{B}_n$, can be set arbitrarily.
- $\mathsf{EncS}'(j, y)$. Run $\mathsf{EncS}^{(j)}(y)$ to obtain $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}$. Let $\mathbf{s}_{\mathrm{new}} = (\mathbf{s}_0[1])\mathbf{1}_1^{d_2}$.
- $\mathsf{EncR}'((i, x), (j, y))$. Set $\mathbf{a}_{\mathrm{new}} = \mathbf{1}_1^{d_2}$, $\mathbf{r}_{\mathrm{new}} = -\mathbf{1}_1^{d_1}$. Since $\bar{P}_\kappa\big((i, x), (j, y)\big) = 0$, we have two cases:
  - Case I: $i = j$ and $P_{\kappa_j}^{(j)}(x, y) = 0$. Run $\mathsf{EncR}^{(j)}(x, y)$ to obtain all the vectors $\mathbf{r}_v, \hat{\mathbf{r}}_u$.
  - Case II: $i \neq j$. Set all the vectors $\mathbf{r}_v, \hat{\mathbf{r}}_u$ to 0.

We verify the selective symbolic property as follows. First, we have $\mathbf{a}_{\mathrm{new}}\mathbf{s}_{\mathrm{new}}^\top = (\mathbf{s}_0[1])\mathbf{1}_1^{d_2}(\mathbf{1}_1^{d_2})^\top = \mathbf{a}\mathbf{s}_0^\top \neq 0$ due to $\mathsf{Sym\text{-}Prop}$ (P1) of $\Gamma^{(j)}$.

For $\mathsf{ct\text{-}enc}$, we have that all polynomials in $\mathbf{c}$ are substituted and evaluated to 0 thanks to $\mathsf{Sym\text{-}Prop}$ (P2) of $\Gamma^{(j)}$. The polynomial $g_j s_0 + h_j s_{\mathrm{new}}$ is substituted and evaluated to

$$\mathbf{G}_j\mathbf{s}_0^\top + \mathbf{H}_j\mathbf{s}_{\mathrm{new}}^\top = -\mathbf{1}_{1,1}^{d_1\times d_2}\mathbf{s}_0^\top + \mathbf{1}_{1,1}^{d_1\times d_2}(\mathbf{s}_0[1])(\mathbf{1}_1^{d_2})^\top = \begin{pmatrix} -\mathbf{s}_0[1] + \mathbf{s}_0[1] \\ 0 \end{pmatrix} = 0. \tag{36}$$

For $\mathsf{key\text{-}enc}$, the polynomial $\alpha_{\mathrm{new}} + r_{\mathrm{new}} h_i$ is substituted and evaluated to $\mathbf{a}_{\mathrm{new}} + \mathbf{r}_{\mathrm{new}}\mathbf{H}_i = \mathbf{1}_1^{d_2} - \mathbf{1}_1^{d_1}\mathbf{1}_{1,1}^{d_1\times d_2} = 0$. For the polynomials in $\tilde{\mathbf{k}}$, consider the above two cases.

- Case I ($i = j$). Since $\mathbf{r}_{\text{new}}\mathbf{G}_j = (-\mathbf{1}_1^{d_1})(-\mathbf{1}_{1,1}^{d_1 \times d_2}) = \mathbf{a}$, the substitution for $\tilde{\mathbf{k}}$ becomes exactly the same as that of $\mathbf{k}$ via $\mathsf{EncR}^{(j)}$, where the polynomials are substituted and evaluated to 0 thanks to Sym-Prop (P2) of $\Gamma^{(j)}$.
- Case II ($i \neq j$). These polynomials are linear combinations $r_{\text{new}}g_i$, $r_v b_\iota$, $\hat{r}_u$, which are substituted to $\mathbf{r}_{\text{new}}\mathbf{G}_i$, $\mathbf{r}_v\mathbf{B}_\iota$, $\hat{\mathbf{r}}_u$, respectively, which are all evaluated to 0, since $\mathbf{G}_i = 0$, $\mathbf{r}_v = 0$, $\hat{\mathbf{r}}_u = 0$.

Next, Turning to prove co-selective symbolic property, we define the following.

- $\mathsf{EncB}'(i, x)$. Run $\mathsf{EncB}^{(i)}(x)$ to obtain $\mathbf{B}_1, \ldots, \mathbf{B}_{n_i}$. Set $\mathbf{G}_i = -\mathbf{1}_{1,1}^{d_1 \times d_2}$. Set $\mathbf{H}_i = \mathbf{1}_{1,1}^{d_1 \times d_2}$. Set $\mathbf{H}_1, \ldots, \mathbf{H}_{i-1}, \mathbf{H}_{i+1} \ldots, \mathbf{H}_k$ to 0. The remaining matrices, $\mathbf{G}_1, \ldots, \mathbf{G}_{j-1}, \mathbf{G}_{j+1}, \ldots, \mathbf{G}_k$ and $\mathbf{B}_{n_i+1}, \ldots, \mathbf{B}_n$, can be set arbitrarily.
- $\mathsf{EncR}'(i, x)$. Run $\mathsf{EncR}^{(i)}(x)$ to obtain all the vectors $\mathbf{r}_v, \hat{\mathbf{r}}_u$. Set $\mathbf{a}_{\text{new}} = \mathbf{1}_1^{d_2}$, and $\mathbf{r}_{\text{new}} = -\mathbf{1}_1^{d_1}$.
- $\mathsf{EncS}'((i, x), (j, y))$. Since $\bar{P}_\kappa\big((i, x),\ (j, y)\big) = 0$, we have two cases:
  - Case I: $j = i$ and $P_{\kappa_i}^{(i)}(x, y) = 0$. Run $\mathsf{EncS}^{(i)}(x, y)$ to obtain $\mathbf{s}_0, \ldots, \mathbf{s}_{w_1}$, $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}$. Let $\mathbf{s}_{\text{new}} = (\mathbf{s}_0[1])\mathbf{1}_1^{d_2}$.
  - Case II: $j \neq i$. Let $\mathbf{s}_{\text{new}} = \mathbf{1}_1^{d_2}$, and set all the other vectors, $\mathbf{s}_t, \hat{\mathbf{s}}_z$, to 0.

We verify the co-selective symbolic property as follows. First, it is straightforward to see that $\mathbf{a}_{\text{new}}\mathbf{s}_{\text{new}}^\top \neq 0$ for both cases of $\mathbf{s}_{\text{new}}$.

For key-enc, the polynomial $\alpha_{\text{new}} + r_{\text{new}}h_i$ is substituted and evaluated to $\mathbf{a}_{\text{new}} + \mathbf{r}_{\text{new}}\mathbf{H}_i = \mathbf{1}_1^{d_2} - \mathbf{1}_1^{d_1}\mathbf{1}_{1,1}^{d_1 \times d_2} = 0$. The polynomials in $\tilde{\mathbf{k}}$ are substituted and evaluated to 0 thanks to Sym-Prop (P2) of $\Gamma^{(i)}$ and the fact that $\mathbf{r}_{\text{new}}\mathbf{G}_i = (-\mathbf{1}_1^{d_1})(-\mathbf{1}_{1,1}^{d_1 \times d_2}) = \mathbf{1}_1^{d_2} = \mathbf{a}$.

For ct-enc, consider the above two cases.

- Case I ($j = i$). The polynomials in $\mathbf{c}$ are are substituted and evaluated to 0 thanks to Sym-Prop (P2) of $\Gamma^{(i)}$. The polynomial $g_j s_0 + h_j s_{\text{new}}$ is substituted and evaluated to 0, exactly as in Eq.(36).
- Case II ($j \neq i$). The polynomials in $\mathbf{c}$ are linear combinations of $b_\iota s_t$ and $\hat{s}_z$, which are substituted to $\mathbf{B}_\iota\mathbf{s}_t^\top$ and $\hat{\mathbf{s}}_z^\top$, respectively, all of which are all evaluated to 0, since $\mathbf{s}_t = 0$, $\hat{\mathbf{s}}_z = 0$. The polynomial $g_j s_0 + h_j s_{\text{new}}$ is substituted and evaluated to

$$\mathbf{G}_j\mathbf{s}_0^\top + \mathbf{H}_j\mathbf{s}_{\text{new}}^\top = \mathbf{G}_j 0 + 0(\mathbf{1}_1^{d_2})^\top = 0.$$

This concludes the proof. □

I Proof for Predicative Automata

This section provides the symbolic security proof for Construction 6.

I.1 Properties for Predicative DFA

We first state useful propositions for general properties of predicative DFA. Notably, Proposition 4 below provides a necessary combinatorial condition for (M, Y) when a predicative DFA machine M does not accept an input Y. These properties are general and are not specific to our scheme. They will be used to define the "mask" vectors in the proof.

Notation and Some Properties. We generalize some notations and properties for DFA given in [41,7,2] to predicative automata. Fixing a predicative automata $M = (Q, \mathcal{T}, q_0, q_{\sigma-1})$ over predicate P_κ, with $Q = \{q_0, \dots, q_{\sigma-1}\}$ and $\mathcal{T} = \left\{ (q_{\upsilon_t}, q_{\omega_t}, x_t) \right\}_{t\in[m]}$, and fixing an input $Y = (y_1, \dots, y_\ell) \in (\mathcal{Y}_\kappa)^*$, we define some notations as follows. For $i \in [0, \ell]$, let $Y_i := (y_{i+1}, \dots, y_\ell)$, that is, the vector formed by the last $\ell - i$ elements of Y. Thus, $Y_0 = Y$ and Y_ℓ is empty. For $k \in [0, \sigma-1]$, let M_k be the same predicative automata as M except that the start state is set to q_k. For $k \in [0, \sigma-1]$, we define

$$V_k := \{\, i \in [0, \ell] \mid M_k \text{ accepts } Y_i \,\}.$$

We also let $V_k^{+1} := \{\, i+1 \mid i \in V_k \,\}$. Hence, $V_k^{+1} \subseteq [1, \ell+1]$. Conversely, for $i \in [0, \ell]$, we define

$$U_i := \{\, k \in [0, \sigma-1] \mid M_k \text{ accepts } Y_i \,\}.$$

Proposition 4. *For any M, Y, we have the following.*

1. *$\ell \notin V_k$ for all $k \in [0, \sigma-2]$.*
2. *$V_{\sigma-1} = \{\ell\}$ and $U_\ell = \{\sigma-1\}$.*

Proof. First, for $k \in [0, \sigma-2]$, M_k starts with a non-accept state (since we have only one accept state, $q_{\sigma-1}$). Hence, V_k for such k does not accept an empty string Y_ℓ; therefore, $\ell \notin V_k$. Next, since $M_{\sigma-1}$ starts with the accept state, which has no outgoing transition, it always accepts only an empty string Y_ℓ, hence we have $V_{\sigma-1} = \{\ell\}$ and $U_\ell = \{\sigma-1\}$. □

Proposition 5. *Suppose that M does* not *accept Y. Then, we have the following.*

1. *$0 \notin V_0$ and $0 \notin U_0$.*
2. *For $t \in [1, m]$, $i \in [1, \ell]$, we have*

$$i \in (V_{\upsilon_t}^{+1} \setminus V_{\omega_t}) \cup (V_{\omega_t} \setminus V_{\upsilon_t}^{+1}) \implies P_\kappa(x_t, y_i) = 0, \tag{37}$$

which is also equivalent to the following:

$$P_\kappa(x_t, y_i) = 1 \implies (\upsilon_t \in U_{i-1} \wedge \omega_t \in U_i) \vee (\upsilon_t \notin U_{i-1} \wedge \omega_t \notin U_i). \tag{38}$$

Proof. Suppose that M does not accept Y. First, since $M_0 = M$ does not accept $Y_0 = Y$, we have $0 \notin V_0$ and $0 \notin U_0$.

For Statement (37), we have two cases:

- Case $i \in (V^{+1}_{\upsilon_t} \setminus V_{\omega_t})$. That is, $i-1 \in V_{\upsilon_t}$ but $i \notin V_{\omega_t}$. First, $i-1 \in V_{\upsilon_t}$ implies that M_{υ_t} accepts $Y_{i-1} = (y_i, \ldots, y_\ell)$. Suppose $P_\kappa(x_t, y_i) = 1$. Due to the determinism of M, the transition $(\upsilon_t, \omega_t, x_t) \in \mathfrak{T}$ implies that M_{ω_t} would accept $(y_{i+1}, \ldots, y_\ell) = Y_i$. Thus, $i \in V_{\omega_t}$, a contradiction. Hence, it must be that $P_\kappa(x_t, y_i) = 0$. Note that without determinism, it might *not* hold that M_{ω_t} accepts Y_i (when assuming $P_\kappa(x_t, y_i) = 1$). This is since there might be another transition $(\upsilon_{t'}, \omega_{t'}, x_{t'}) \in \mathfrak{T}$ with $\upsilon_{t'} = \upsilon_t$ and $P_\kappa(x_{t'}, y_i) = 1$ but $\omega_{t'} \neq \omega_t$, and we might have $M_{\omega_{t'}}$ accepts Y_i instead.
- Case $i \in (V_{\omega_t} \setminus V^{+1}_{\upsilon_t})$. That is, $i \in V_{\omega_t}$ but $i-1 \notin V_{\upsilon_t}$. First, $i \in V_{\omega_t}$ implies that M_{ω_t} accepts $Y_i = (y_{i+1}, \ldots, y_\ell)$. Suppose $P_\kappa(x_t, y_i) = 1$. The transition $(\upsilon_t, \omega_t, x_t) \in \mathfrak{T}$ implies that M_{υ_t} would accept $(y_i, y_{i+1}, \ldots, y_\ell) = Y_{i-1}$. Thus, $i-1 \in V_{\upsilon_t}$, a contradiction. Hence, it must be that $P_\kappa(x_t, y_i) = 0$.

Statement (38) is merely the contrapositive of Statement (37). Note that $i \notin (V^{+1}_{\upsilon_t} \setminus V_{\omega_t}) \cup (V_{\omega_t} \setminus V^{+1}_{\upsilon_t})$ implies the remaining two cases: $i-1 \in V_{\upsilon_t} \wedge i \in V_{\omega_t}$, or $i-1 \notin V_{\upsilon_t} \wedge i \notin V_{\omega_t}$. That is, $\upsilon_t \in U_{i-1} \wedge \omega_t \in U_i$ or $\upsilon_t \notin U_{i-1} \wedge \omega_t \notin U_i$. □

I.2 Summary for Polynomials/Variables in Our Construction

We list polynomials and variables of Construction 6, for referring in the proof.

- A ciphertext encoding $\mathbf{c}' = \mathbf{c}'(\mathbf{s}', \hat{\mathbf{s}}', \mathbf{b}')$ is

$$\mathbf{c}' = \big(c'_0, c'_1, \ldots, c'_\ell, \big(\mathbf{c}^{(1,i)}, \mathbf{c}^{(2,i)} \big)_{i \in [\ell]} \big),$$

in variable $\mathbf{b}'$ and

$$\mathbf{s}' := \big(s^{(\ell)}_{\text{new}}, s^{(0)}_{\text{new}}, \ldots, s^{(\ell-1)}_{\text{new}}, \big(\mathbf{s}^{(1,i)}, \mathbf{s}^{(2,i)} \big)_{i \in [\ell]} \big),$$
$$\hat{\mathbf{s}}' := \big(\hat{\mathbf{s}}^{(1,i)}, \hat{\mathbf{s}}^{(2,i)} \big)_{i \in [\ell]}.$$

Note that $\mathbf{c}^{(1,i)} = \mathbf{c}^{(i)}(\mathbf{s}^{(1,i)}, \hat{\mathbf{s}}^{(1,i)}, \mathbf{b}_1)$ contains variables $\mathbf{b}_1$ and

$$\mathbf{s}^{(1,i)} = (s^{(1,i)}_0, \ldots, s^{(1,i)}_{w_{1,i}}), \qquad \hat{\mathbf{s}}^{(1,i)} = (\hat{s}^{(1,i)}_1, \ldots, \hat{s}^{(1,i)}_{w_{2,i}}).$$

Also, $\mathbf{c}^{(2,i)} = \mathbf{c}^{(i)}(\mathbf{s}^{(2,i)}, \hat{\mathbf{s}}^{(2,i)}, \mathbf{b}_2)$ contains variables $\mathbf{b}_2$ and

$$\mathbf{s}^{(2,i)} = (s^{(2,i)}_0, \ldots, s^{(2,i)}_{w_{1,i}}) \qquad \hat{\mathbf{s}}^{(2,i)} = (\hat{s}^{(2,i)}_1, \ldots, \hat{s}^{(2,i)}_{w_{2,i}}).$$

Note that the two sets of variables are related via Eq.(18).
- A key encoding $\mathbf{k}' = \mathbf{k}'(\mathbf{r}', \hat{\mathbf{r}}', \mathbf{b}')$ is

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

in variable $\mathbf{b}'$ and

$$\mathbf{r}' := \Big(r^{(0)}_{\text{new}}, \ldots, r^{(m)}_{\text{new}}, \big(\mathbf{r}^{(1,t)}, \mathbf{r}^{(2,t)} \big)_{t \in [m]} \Big),$$
$$\hat{\mathbf{r}}' := \Big(u_{\sigma-1}, u_0, u_1, \ldots, u_{\sigma-2}, \big(\hat{\mathbf{r}}'^{(1,t)}, \hat{\mathbf{r}}'^{(2,t)} \big)_{t \in [m]} \Big),$$

where $\mathbf{k}'^{(1,t)}$ contains variables

$$\mathbf{r}^{(1,t)} = (r_1^{(1,t)}, \ldots, r_{m_{1,t}}^{(1,t)}) \qquad \hat{\mathbf{r}}'^{(1,t)} := (\hat{r}_1^{(1,t)}, \ldots, \hat{r}_{m_{2,t}}^{(1,t)}),$$

and $\mathbf{k}'^{(2,t)}$ contains variables

$$\mathbf{r}^{(2,t)} = (r_1^{(2,t)}, \ldots, r_{m_{1,t}}^{(2,t)}) \qquad \hat{\mathbf{r}}'^{(2,t)} := (\hat{r}_1^{(2,t)}, \ldots, \hat{r}_{m_{2,t}}^{(2,t)}),$$

Note that, here we newly define $\hat{\mathbf{r}}'^{(1,t)}$ to be exactly $\hat{\mathbf{r}}^{(1,t)}$ but without $\alpha^{(1,t)}$. The same goes for the second set.

I.3 Proof for Our Predicative DFA Scheme

Disclaimer. We opted to write matrices/vectors in their full expanded forms for better visualization. This somewhat generates seemingly a longer proof than it should be. The mechanism inside is actually fairly simple and directly reflects the combinatorial properties of the non-acceptance conditions for predicate DFA. Moreover, since verification of symbolic properties involves only doing simple linear algebra (namely, matrix multiplications), the proof below is actually fairly easy to verify.

Selective Symbolic Property. We first prove selective symbolic property of Γ' from that of Γ. We will use $\mathsf{Sym\text{-}Prop}^{++}$ (*cf.* Definition 8), where we consider a fixed dummy attribute $y_0 = \perp$, where $P_\kappa(x, y_0) = 0$ for any x. We define the following algorithms.

- $\mathsf{EncBS}'(Y)$. Parse $Y = (y_1, \ldots, y_\ell)$. Proceed as follows.
  1. For each $i \in [\ell]^+$, run
  $$\mathsf{EncB}(y_i) \to \Big(\mathbf{B}_1^{(i)}, \ldots, \mathbf{B}_n^{(i)};\ \Big),$$
  and for each $i \in [\ell]$, run
  $$\mathsf{EncS}(y_i) \to \Big(\mathbf{s}_0^{(i)}, \ldots, \mathbf{s}_{w_{1,i}}^{(i)};\ \hat{\mathbf{s}}_1^{(i)}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{(i)}\Big),$$
  where $\mathbf{B}_j^{(i)} \in \mathbb{Z}_N^{d_1 \times d_2}$, $\mathbf{s}_\tau^{(i)} \in \mathbb{Z}_N^{1 \times d_2}$, $\hat{\mathbf{s}}_z^{(i)} \in \mathbb{Z}_N^{1 \times d_1}$. Let $d_1' = (\ell+1)d_1$ and $d_2' = (\ell+1)d_2$. Note that $\mathsf{EncB}(y_0 = \perp)$ is available due to $\mathsf{Sym\text{-}Prop}^{++}$.
  2. For $j \in [n]$, let $\mathbf{B}_{1,j} = 0$ and
  $$\mathbf{B}_{2,j} = \begin{pmatrix} \mathbf{B}_j^{(0)} & & & \\ & \mathbf{B}_j^{(1)} & & \\ & & \ddots & \\ & & & \mathbf{B}_j^{(\ell)} \end{pmatrix} \in \mathbb{Z}_N^{d_1' \times d_2'}.$$

3. Let $\mathbf{G}_1 = 0$ and

$$\mathbf{H}_1 = \begin{array}{r@{\;}c}
 & \begin{array}{ccccc} 0 & 1 & & \ell-1 & \ell \end{array} \\
\begin{array}{r} 1 \\ 2 \\ \\ \ell \\ \\ \end{array} &
\left(\begin{array}{cccc|c}
\mathbf{a} & & & & \\
 & \mathbf{a} & & & 0 \\
 & & \ddots & & \\
 & & & \mathbf{a} & \\ \hline
\multicolumn{5}{c}{0}
\end{array}\right)
\end{array}, \qquad
\mathbf{H}_0 = \begin{array}{r@{\;}c}
 & \begin{array}{ccccc} 0 & 1 & & & \ell \end{array} \\
\begin{array}{r} 1 \\ 2 \\ \\ \ell \\ \\ \end{array} &
\left(\begin{array}{c|cccc}
 & \mathbf{a} & & & \\
0 & & \mathbf{a} & & \\
 & & & \ddots & \\
 & & & & \mathbf{a} \\ \hline
\multicolumn{5}{c}{0}
\end{array}\right)
\end{array},$$

and let

$$\mathbf{H}_2 = -\begin{array}{r@{\;}c}
 & \begin{array}{ccccc} 0 & 1 & & & \ell \end{array} \\
\begin{array}{r} 1 \\ 2 \\ \\ \ell \\ \ell+1 \\ \ell+2 \\ \\ 2\ell \\ 2\ell+1 \\ 2\ell+2 \\ 2\ell+3 \\ \\ 3\ell+1 \\ \\ \end{array} &
\left(\begin{array}{c|cccc}
 & \mathbf{a} & & & \\
0 & & \mathbf{a} & & \\
 & & & \ddots & \\
 & & & & \mathbf{a} \\ \hline
 & \mathbf{a} & & & \\
0 & & \mathbf{a} & & \\
 & & & \ddots & \\
 & & & & \mathbf{a} \\ \hline
\mathbf{a} & & & & \\
 & \mathbf{a} & & & \\
 & & \mathbf{a} & & \\
 & & & \ddots & \\
 & & & & \mathbf{a} \\ \hline
\multicolumn{5}{c}{0}
\end{array}\right)
\end{array}, \qquad
\mathbf{G}_2 = \begin{array}{r@{\;}c}
 & \begin{array}{ccccc} 0 & 1 & & & \ell \end{array} \\
\begin{array}{r} 1 \\ 2 \\ \vdots \\ \ell \\ \ell+1 \\ \ell+2 \\ \\ 2\ell \\ 2\ell+1 \\ 2\ell+2 \\ 2\ell+3 \\ \\ 3\ell+1 \\ \\ \end{array} &
\left(\begin{array}{c|cccc}
\multicolumn{5}{c}{} \\
\multicolumn{5}{c}{0} \\
\multicolumn{5}{c}{} \\
\multicolumn{5}{c}{} \\ \hline
 & \mathbf{a} & & & \\
0 & & \mathbf{a} & & \\
 & & & \ddots & \\
 & & & & \mathbf{a} \\ \hline
\mathbf{a} & & & & \\
 & \mathbf{a} & & & \\
 & & \mathbf{a} & & \\
 & & & \ddots & \\
 & & & & \mathbf{a} \\ \hline
\multicolumn{5}{c}{0}
\end{array}\right)
\end{array}$$

$\in \mathbb{Z}_N^{d_1' \times d_2'}$, and recall that $\mathbf{a} = \mathbf{1}_1^{d_2}$.

4. For $i \in [\ell]^+$, let

$$\mathbf{s}_{\text{new}}^{(i)} = (0, \ldots, 0, \overset{\substack{\text{block } i \\ \downarrow}}{\mathbf{1}_1^{d_2}}, 0, \ldots, 0) = \mathbf{1}_{id_2+1}^{d_2'} \in \mathbb{Z}_N^{1 \times d_2'},$$

where the vector of length $d_2' = (\ell+1)d_2$ is divided to $(\ell+1)$ blocks of length d_2, starting from block 0 to block ℓ.[20] Blocks in d_1'-length vectors are defined similarly.

5. For $i \in [\ell]$, compute $\rho_i = 1/\big(\mathbf{s}_0^{(i)}[1]\big)$, which is computable since $\mathbf{s}_0^{(i)}[1] = \mathbf{a}(\mathbf{s}_0^{(i)})^\top \neq 0$ (from the symbolic property of $\mathit{\Gamma}$).

[20] Hence, here, block i contains elements of position $id_2 + 1$ to $(i+1)d_2$.

6. For $i \in [\ell]$, $\tau \in [w_{1,i}]^+$, $z \in [w_{2,i}]$, let

$$\mathbf{s}_\tau^{(2,i)} = (0, \ldots, 0, \overset{\substack{\text{block } i \\ \downarrow}}{\rho_i \mathbf{s}_\tau^{(i)}}, 0, \ldots, 0) \in \mathbb{Z}_N^{1 \times d_2'},$$

$$\hat{\mathbf{s}}_z^{(2,i)} = (0, \ldots, 0, \overset{\substack{\text{block } i \\ \downarrow}}{\rho_i \hat{\mathbf{s}}_z^{(i)}}, 0, \ldots, 0) \in \mathbb{Z}_N^{1 \times d_1'},$$

   and $\hat{\mathbf{s}}_z^{(1,i)} = 0$, while $\mathbf{s}_\tau^{(1,i)}$ is defined arbitrarily with only the constraint being Eq.(18).
7. Finally, output

$$\begin{aligned}
\mathbb{B}' &= \left(\mathbf{B}_{1,1}, \ldots, \mathbf{B}_{1,n}, \mathbf{B}_{2,1}, \ldots, \mathbf{B}_{2,n}, \mathbf{H}_0, \mathbf{G}_1, \mathbf{H}_1, \mathbf{G}_2, \mathbf{H}_2\right), \\
\mathbb{S}' &= \left(\mathbf{s}_{\text{new}}^{(\ell)}, \mathbf{s}_{\text{new}}^{(0)}, \ldots, \mathbf{s}_{\text{new}}^{(\ell-1)}, \left(\mathbf{s}_0^{(1,i)}, \ldots, \mathbf{s}_{w_{1,i}}^{(1,i)}, \mathbf{s}_0^{(2,i)}, \ldots, \mathbf{s}_{w_{1,i}}^{(2,i)}\right)_{i \in [\ell]}\right), \\
\hat{\mathbb{S}}' &= \left(\hat{\mathbf{s}}_1^{(1,i)}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{(1,i)}, \hat{\mathbf{s}}_1^{(2,i)}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{(2,i)}\right)_{i \in [\ell]}
\end{aligned}$$

   which define substitutions $\mathbf{b}' : \mathbb{B}'$, $\mathbf{s}' : \mathbb{S}'$, and $\hat{\mathbf{s}}' : \hat{\mathbb{S}}'$.

– $\mathsf{EncR}'(M, Y)$. First note that we have the condition $\bar{P}_\kappa(M, Y) = 0$.
1. Parse $M = (Q = \{q_0, \ldots, q_{\sigma-1}\}, \mathcal{T} = \{ (q_{v_t}, q_{\omega_t}, x_t) \}_{t \in [m]}, q_0, q_{\sigma-1})$. Parse $Y = (y_1, \ldots, y_\ell)$.
2. For $t \in [m], i \in [\ell]^+$ such that $P_\kappa(x_t, y_i) = 0$, it is possible to run[21]

$$\mathsf{EncR}(x_t, y_i) \to \left(\mathbf{r}_1^{\langle t,i \rangle}, \ldots, \mathbf{r}_{m_{1,t}}^{\langle t,i \rangle};\ \mathbf{a}, \hat{\mathbf{r}}_1^{\langle t,i \rangle}, \ldots, \hat{\mathbf{r}}_{m_{2,t}}^{\langle t,i \rangle}\right).$$

   where $\mathbf{r}_v^{\langle t,i \rangle} \in \mathbb{Z}_N^{1 \times d_1}$, $\hat{\mathbf{r}}_\mu^{\langle t,i \rangle} \in \mathbb{Z}_N^{1 \times d_2}$, and $\mathbf{a} = \mathbf{1}_1^{d_2} \in \mathbb{Z}_N^{1 \times d_2}$. Note that, for $i = 0$ we have $y_0 = \perp$, and $\mathsf{EncR}(x_t, \perp)$ is available due to $\mathsf{Sym\text{-}Prop}^{++}$.
3. For $t \in [m], i \in [\ell]^+$, let[22]

$$\rho^{\langle t,i \rangle} := \begin{cases} -1 & \text{if } i \in V_{v_t}^{+1} \setminus V_{\omega_t} \\ 1 & \text{if } i \in V_{\omega_t} \setminus V_{v_t}^{+1} \\ 0 & \text{otherwise} \end{cases}. \tag{39}$$

4. Let $\mathbf{r}_v^{(1,t)} = 0$, $\hat{\mathbf{r}}_\mu^{(1,t)} = 0$, and

$$\mathbf{r}_v^{(2,t)} = (\rho^{\langle t,0 \rangle} \mathbf{r}_v^{\langle t,0 \rangle}, \ldots, \rho^{\langle t,\ell \rangle} \mathbf{r}_v^{\langle t,\ell \rangle}) \in \mathbb{Z}_N^{1 \times d_1'}, \tag{40}$$

$$\hat{\mathbf{r}}_\mu^{(2,t)} = (\rho^{\langle t,0 \rangle} \hat{\mathbf{r}}_\mu^{\langle t,0 \rangle}, \ldots, \rho^{\langle t,\ell \rangle} \hat{\mathbf{r}}_\mu^{\langle t,\ell \rangle}) \in \mathbb{Z}_N^{1 \times d_2'}, \tag{41}$$

   for $v \in [m_{1,t}]$, $\mu \in [m_{2,t}]$.

[21] Note that $\langle t, i \rangle$ denotes a tuple of t and i, and is used for superscript here.

[22] Intuitively, $(\rho^{\langle t,0 \rangle}, \ldots, \rho^{\langle t,\ell \rangle})$ plays the role of the "mask" vector, as motivated at the end of §2: it encodes exactly the non-acceptance condition of DFA M (Proposition 5).

5. For $t \in [m]$, let

$$\mathbf{r}_{\text{new}}^{(t)} = \left(\sum_{i \in V_{\upsilon_t}^{+1}} \mathbf{1}_i^{d'_1} \right) - \left(\sum_{i \in V_{\upsilon_t}^{+1} \setminus V_{\omega_t}} \mathbf{1}_{\ell+i}^{d'_1} \right) + \left(\sum_{i \in V_{\omega_t} \setminus V_{\upsilon_t}^{+1}} \mathbf{1}_{2\ell+1+i}^{d'_1} \right) \in \mathbb{Z}_N^{1 \times d'_1}. \tag{42}$$

Note that since $\ell \notin V_k$ for any $k \in [0, \sigma - 2]$ (*cf.* Proposition 4) and $\upsilon_t \in [0, \sigma - 2]$ (since it is a "from" state of a transition, and the accept state $q_{\sigma-1}$ has no outgoing transition), we have $V_{\upsilon_t} \subseteq [0, \ell - 1]$, that is, $V_{\upsilon_t}^{+1} \subseteq [1, \ell]$. Therefore, the above definition of $\mathbf{r}_{\text{new}}^{(t)}$ is well-defined.[23] Moreover, we have $V_{\upsilon_t}^{+1} \setminus V_{\omega_t} \subseteq [1, \ell]$ but $V_{\omega_t} \setminus V_{\upsilon_t}^{+1} \subseteq [0, \ell]$. Hence, the index in all the three sums (in $\mathbf{r}_{\text{new}}^{(t)}$) are $[1, \ell]$, $[\ell+1, 2\ell]$, and $[2\ell+1, 3\ell+1]$, which are disjointed.
6. Let

$$\mathbf{r}_{\text{new}}^{(0)} = \sum_{i \in V_0} \mathbf{1}_i^{d'_1} \in \mathbb{Z}_N^{1 \times d'_1}.$$

Note that since $0 \notin V_0$ (due to Proposition 5), this is well-defined.
7. For $k \in [0, \sigma - 1]$, let $\mathbf{u}_k = (\mathbf{u}_{k,0}, \ldots, \mathbf{u}_{k,\ell}) \in \mathbb{Z}_N^{1 \times d'_2}$, where for $i \in [\ell]^+$ each block $\mathbf{u}_{k,i}$ is of length d_2 and

$$\mathbf{u}_{k,i} \coloneqq \begin{cases} -\mathbf{a} & \text{if } i \in V_k \\ 0 & \text{if } i \notin V_k \end{cases}.$$

That is, $\mathbf{u}_k = -\left(\sum_{i \in V_k} \mathbf{1}_{id_2+1}^{d'_2} \right)$.
8. Finally, output

$$\mathbb{R}' = \left(\mathbf{r}_{\text{new}}^{(0)}, \ldots, \mathbf{r}_{\text{new}}^{(m)}, \left(\mathbf{r}_1^{(1,t)}, \ldots, \mathbf{r}_{m_{1,t}}^{(1,t)}, \mathbf{r}_1^{(2,t)}, \ldots, \mathbf{r}_{m_{1,t}}^{(2,t)} \right)_{t \in [m]} \right),$$
$$\hat{\mathbb{R}}' = \left(\mathbf{u}_{\sigma-1}, \mathbf{u}_0, \ldots, \mathbf{u}_{\sigma-2}, \left(\hat{\mathbf{r}}_1^{(1,t)}, \ldots, \hat{\mathbf{r}}_{m_{2,t}}^{(1,t)}, \hat{\mathbf{r}}_1^{(2,t)}, \ldots, \hat{\mathbf{r}}_{m_{2,t}}^{(2,t)} \right)_{t \in [m]} \right)$$

which define substitutions $\mathbf{r}' : \mathbb{R}'$ and $\hat{\mathbf{r}}' : \mathbb{R}'$.

Verifying Properties. Since $V_{\sigma-1} = \{\ell\}$ (*cf.* Proposition 4), we have $\mathbf{u}_{\sigma-1} = -\mathbf{1}_{\ell d_2+1}^{d'_2}$. We can thus verify that,

$$\mathbf{u}_{\sigma-1} (\mathbf{s}_{\text{new}}^{(\ell)})^\top = -\mathbf{1}_{\ell d_2+1}^{d'_2} (\mathbf{1}_{\ell d_2+1}^{d'_2})^\top = -1$$

which is not zero, as required.

For ct-enc, we have the following polynomials.

[23] In particular, there is no such $\mathbf{1}_0^{d'_1}$, which is undefined.

- The polynomial $c'_0 = h_0 s^{(0)}_{\text{new}}$ is substituted and evaluated to

$$\mathbf{H}_0\mathbf{s}^{(0)}_{\text{new}} = \left(\begin{array}{c|c|cccc}
 & 0 & 1 & & & \ell \\ \hline
1 & & \mathbf{a} & & & \\
2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
\ell & & & & & \mathbf{a} \\ \hline
 & \multicolumn{5}{c}{0}
\end{array}\right)(\mathbf{1}_1^{d'_2})^\top = 0.$$

- For $i \in [\ell]$, the polynomial $c'_i = h_1 s^{(i-1)}_{\text{new}} + g_1 s'^{(i-1)}_0 + h_2 s^{(i)}_{\text{new}} + g_2 s'^{(i)}_0$ is substituted and evaluated as follows. For the first term, $h_1 s^{(i-1)}_{\text{new}}$, it is

$$\mathbf{H}_1(\mathbf{s}^{(i-1)}_{\text{new}})^\top = \left(\begin{array}{c|cccc|c}
 & 0 & 1 & & \ell-1 & \ell \\ \hline
1 & \mathbf{a} & & & & \\
2 & & \mathbf{a} & & & 0 \\
 & & & \ddots & & \\
\ell & & & & \mathbf{a} & \\ \hline
 & \multicolumn{5}{c}{0}
\end{array}\right)(0,\dots,0,\ \overset{\substack{\text{block } i-1\\ \downarrow}}{\mathbf{1}_1^{d_2}}\ ,0,\dots,0)^\top = (\mathbf{1}_i^{d'_1})^\top.$$

The second term, $g_1 s'^{(i-1)}_0$, is substituted and evaluated to 0, since $\mathbf{G}_1 = 0$. The third term, $h_2 s^{(i)}_{\text{new}}$, is substituted and evaluated to

$$\mathbf{H}_2(\mathbf{s}^{(i)}_{\text{new}})^\top = -\left(\begin{array}{c|c|cccc}
 & 0 & 1 & & & \ell \\ \hline
1 & & \mathbf{a} & & & \\
2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
\ell & & & & & \mathbf{a} \\ \hline
\ell+1 & & \mathbf{a} & & & \\
\ell+2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
2\ell & & & & & \mathbf{a} \\ \hline
2\ell+1 & \mathbf{a} & & & & \\
2\ell+2 & & \mathbf{a} & & & \\
2\ell+3 & & & \mathbf{a} & & \\
 & & & & \ddots & \\
3\ell+1 & & & & & \mathbf{a} \\ \hline
 & \multicolumn{5}{c}{0}
\end{array}\right)(0,\dots,0,\ \overset{\substack{\text{block } i\\ \downarrow}}{\mathbf{1}_1^{d_2}}\ ,0,\dots,0)^\top$$

$$= -(\mathbf{1}_i^{d'_1})^\top - (\mathbf{1}_{\ell+i}^{d'_1})^\top - (\mathbf{1}_{2\ell+1+i}^{d'_1})^\top.$$

The fourth term, $g_2 s_0'^{(i)}$, is substituted and evaluated to

$$\mathbf{G}_2(\mathbf{s}_0'^{(i)})^\top = \begin{array}{c|c|cccc|}
 & 0 & 1 & & & \ell \\ \hline
1 & & & & & \\
2 & \multicolumn{5}{c|}{0} \\
\vdots & & & & & \\
\ell & & & & & \\ \hline
\ell+1 & & \mathbf{a} & & & \\
\ell+2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
2\ell & & & & & \mathbf{a} \\ \hline
2\ell+1 & \mathbf{a} & & & & \\
2\ell+2 & & \mathbf{a} & & & \\
2\ell+3 & & & \mathbf{a} & & \\
 & & & & \ddots & \\
3\ell+1 & & & & & \mathbf{a} \\ \hline
 & \multicolumn{5}{c|}{0}
\end{array} (0, \ldots, 0, \overset{\substack{\text{block } i \\ \downarrow}}{\rho_i \mathbf{s}_0^{(i)}}, 0, \ldots, 0)^\top$$

$$= (\mathbf{1}_{\ell+i}^{d_1'})^\top + (\mathbf{1}_{2\ell+1+i}^{d_1'})^\top,$$

where we use that fact that $\rho_i \mathbf{a}(\mathbf{s}_0^{(i)})^\top = 1$. Combining all the four terms, we have $(\mathbf{1}_i^{d_1'})^\top - (\mathbf{1}_i^{d_1'})^\top - (\mathbf{1}_{\ell+i}^{d_1'})^\top - (\mathbf{1}_{2\ell+1+i}^{d_1'})^\top + (\mathbf{1}_{\ell+i}^{d_1'})^\top + (\mathbf{1}_{2\ell+1+i}^{d_1'})^\top = 0$.

– For $i \in [\ell]$, $p \in [w_{3,i}]$, the p-th polynomial in $\mathbf{c}^{(1,i)}$ is

$$\sum_{z \in [w_{2,i}]} \eta_{p,z}^{(i)} \hat{s}_z^{(1,i)} + \sum_{\tau \in [w_{1,i}]^+, j \in [n]} \eta_{p,t,j}^{(i)} b_{1,j} s_\tau^{(1,i)},$$

where we recall that the coefficients are those of $\mathbf{c}^{(i)}$ obtained from $\mathsf{EncCt}(y_i, N)$. We also note that $w_{3,i}$ is the size of $\mathbf{c}^{(i)}$. Via EncBS', it is substituted to

$$\sum_{z \in [w_{2,i}]} \eta_{p,z}^{(i)} (\hat{\mathbf{s}}_z^{(1,i)})^\top + \sum_{\tau \in [w_{1,i}]^+, j \in [n]} \eta_{p,t,j}^{(i)} \mathbf{B}_{1,j} (\mathbf{s}_\tau^{(1,i)})^\top$$

which is evaluated to 0 since $\hat{\mathbf{s}}_z^{(1,i)} = 0$ and $\mathbf{B}_{1,j} = 0$.

– For $i \in [\ell]$, $p \in [w_{3,i}]$, the p-th polynomial in $\mathbf{c}^{(2,i)}$ is

$$\sum_{z \in [w_{2,i}]} \eta_{p,z}^{(i)} \hat{s}_z^{(2,i)} + \sum_{\tau \in [w_{1,i}]^+, j \in [n]} \eta_{p,t,j}^{(i)} b_{2,j} s_\tau^{(2,i)}.$$

Via EncBS', it is substituted

$$\sum_{z \in [w_{2,i}]} \eta_{p,z}^{(i)} (\hat{\mathbf{s}}_z^{(2,i)})^\top + \sum_{\tau \in [w_{1,i}]^+, j \in [n]} \eta_{p,t,j}^{(i)} \mathbf{B}_{2,j} (\mathbf{s}_\tau^{(2,i)})^\top$$

which is evaluated to

$$\sum_{z\in[w_{2,i}]} \eta_{p,z}^{(i)}\ (0,\ldots,0,\ \overset{\substack{\text{block } i\\ \downarrow}}{\rho_i\hat{\mathbf{s}}_z^{(i)}},\qquad\qquad 0,\ldots,0)^\top +$$

$$\sum_{\tau\in[w_{1,i}]^+, j\in[n]} \eta_{p,t,j}^{(i)}(0,\ldots,0,\ \overset{\substack{\text{block } i\\ \downarrow}}{\rho_i(\mathbf{B}_j^{(i)}(\mathbf{s}_\tau^{(i)})^\top)^\top}\ ,0,\ldots,0)^\top$$

which, in turn, is exactly 0, since the sum at the block i is 0 due to the selective symbolic property of Γ. Note also that, in the above, we use

$$\mathbf{B}_{2,j}(\mathbf{s}_\tau^{(2,i)})^\top = \begin{pmatrix} \mathbf{B}_j^{(0)} & & & \\ & \mathbf{B}_j^{(1)} & & \\ & & \ddots & \\ & & & \mathbf{B}_j^{(\ell)} \end{pmatrix} (0,\ldots,0,\overset{\substack{\text{block } i\\ \downarrow}}{\rho_i\mathbf{s}_\tau^{(i)}},0,\ldots,0)^\top$$

$$= (0,\ldots,0,\ \overset{\substack{\text{block } i\\ \downarrow}}{\rho_i(\mathbf{B}_j^{(i)}(\mathbf{s}_\tau^{(i)})^\top)^\top}\ ,0,\ldots,0)^\top.$$

For key-enc, we have the following polynomials.

- The polynomial $\tilde{k}_0 = -u_0 + r_{\text{new}}^{(0)}h_0$ is substituted and evaluated to

$$-\mathbf{u}_0 + \mathbf{r}_{\text{new}}^{(0)}\mathbf{H}_0 = -\left(\sum_{i\in V_0}\mathbf{1}_{id_2+1}^{d_2'}\right) + \left(\sum_{i\in V_0}\mathbf{1}_{i}^{d_1'}\right) \begin{array}{cc} & \begin{array}{cccc} 0 & 1 & & \ell \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \\ \ell \\ \\ \end{array} & \left(\begin{array}{c|cccc} & \mathbf{a} & & & \\ & & \mathbf{a} & & \\ 0 & & & \ddots & \\ & & & & \mathbf{a} \\ \hline & & 0 & & \end{array}\right) \end{array} = 0.$$

- For $t\in[m]$, the polynomial $\tilde{k}_{1,t} = u_{\upsilon_t} + r_{\text{new}}^{(t)}h_1$ is substituted and evaluated as follows. The last term, $r_{\text{new}}^{(t)}h_1$, is substituted to $\mathbf{r}_{\text{new}}^{(t)}\mathbf{H}_1 =$

$$\left(\sum_{i\in V_{\upsilon_t}^{+1}}\mathbf{1}_i^{d_1'} - \sum_{i\in V_{\upsilon_t}^{+1}\setminus V_{\omega_t}}\mathbf{1}_{\ell+i}^{d_1'} + \sum_{i\in V_{\omega_t}\setminus V_{\upsilon_t}^{+1}}\mathbf{1}_{2\ell+1+i}^{d_1'}\right) \begin{array}{cc} & \begin{array}{ccccc} 0 & 1 & & \ell-1 & \ell \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \\ \ell \\ \\ \end{array} & \left(\begin{array}{cccc|c} \mathbf{a} & & & & \\ & \mathbf{a} & & & 0 \\ & & \ddots & & \\ & & & \mathbf{a} & \\ \hline & & 0 & & \end{array}\right) \end{array}$$

$$= \sum_{i\in V_{\upsilon_t}^{+1}}\mathbf{1}_{(i-1)d_2+1}^{d_2'} = \sum_{i\in V_{\upsilon_t}}\mathbf{1}_{id_2+1}^{d_2'}.$$

Now, since $\mathbf{u}_{\upsilon_t} = -\left(\sum_{i\in V_{\upsilon_t}} \mathbf{1}^{d_2'}_{id_2+1}\right)$, we have that $\mathbf{u}_{\upsilon_t} + \mathbf{r}^{(t)}_{\text{new}}\mathbf{H}_1 = 0$.

– For $t \in [m]$, the polynomial $\tilde{k}_{2,t} = -u_{\omega_t} + r^{(t)}_{\text{new}} h_2$ is substituted and evaluated as follows. The last term, $r^{(t)}_{\text{new}} h_2$, is substituted to $\mathbf{r}^{(t)}_{\text{new}}\mathbf{H}_2 =$

$$-\left(\sum_{i\in V^{+1}_{\upsilon_t}} \mathbf{1}^{d_1'}_{i} - \sum_{i\in V^{+1}_{\upsilon_t}\setminus V_{\omega_t}} \mathbf{1}^{d_1'}_{\ell+i} + \sum_{i\in V_{\omega_t}\setminus V^{+1}_{\upsilon_t}} \mathbf{1}^{d_1'}_{2\ell+1+i}\right)
\begin{array}{r|c|cccc|}
 & 0 & 1 & & & \ell \\
1 & & \mathbf{a} & & & \\
2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
\ell & & & & & \mathbf{a} \\ \cline{2-6}
\ell+1 & & \mathbf{a} & & & \\
\ell+2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
2\ell & & & & & \mathbf{a} \\ \cline{2-6}
2\ell+1 & \mathbf{a} & & & & \\
2\ell+2 & & \mathbf{a} & & & \\
2\ell+3 & & & \mathbf{a} & & \\
 & & & & \ddots & \\
3\ell+1 & & & & & \mathbf{a} \\ \cline{2-6}
 & \multicolumn{5}{c|}{0}
\end{array}$$

$$= -\left(\sum_{i\in V_{\omega_t}} \mathbf{1}^{d_2'}_{id_2+1}\right).$$

Now, since $\mathbf{u}_{\omega_t} = -\left(\sum_{i\in V_{\omega_t}} \mathbf{1}^{d_2'}_{id_2+1}\right)$, we have that $-\mathbf{u}_{\omega_t} + \mathbf{r}^{(t)}_{\text{new}}\mathbf{H}_2 = 0$.

– For $t \in [m]$, $p \in [m_{3,t}]$, the p-th polynomial in $\mathbf{k}'^{(1,t)}$ is

$$\phi^{(t)}_p r^{(t)}_{\text{new}} g_1 + \sum_{\mu\in[m_{2,t}]} \phi^{(t)}_{p,\mu} \hat{r}^{(1,t)}_{\mu} + \sum_{v\in[m_{1,t}],j\in[n]} \phi^{(t)}_{p,v,j} r^{(1,t)}_{v} b_{1,j},$$

where we recall that the coefficients are those of $\mathbf{k}^{(t)}$ obtained from $\mathsf{EncKey}(x_t, N)$ (for the PES Γ), and we replace $\alpha^{(1,t)}$ with $r^{(t)}_{\text{new}} g_1$. Also note that $m_{3,t}$ is the size of $\mathbf{k}^{(t)}$. Via EncBS' and EncR', it is substituted to

$$\phi^{(t)}_p \mathbf{r}^{(t)}_{\text{new}} \mathbf{G}_1 + \sum_{\mu\in[m_{2,t}]} \phi^{(t)}_{p,\mu} \hat{\mathbf{r}}^{(1,t)}_{\mu} + \sum_{v\in[m_{1,t}],j\in[n]} \phi^{(t)}_{p,v,j} \mathbf{r}^{(1,t)}_{v} \mathbf{B}_{1,j}$$

but this evaluates to 0, since $\mathbf{G}_1 = 0$, $\hat{\mathbf{r}}^{(1,t)}_{\mu} = 0$, $\mathbf{B}_{1,j} = 0$.

– For $t \in [m]$, $p \in [m_{3,t}]$, the p-th polynomial in $\mathbf{k}'^{(2,t)}$ is

$$\phi^{(t)}_p r^{(t)}_{\text{new}} g_2 + \sum_{\mu\in[m_{2,t}]} \phi^{(t)}_{p,\mu} \hat{r}^{(2,t)}_{\mu} + \sum_{v\in[m_{1,t}],j\in[n]} \phi^{(t)}_{p,v,j} r^{(2,t)}_{v} b_{2,j}.$$

In the first term, $r_{\text{new}}^{(t)} g_2$, evaluates to $\mathbf{r}_{\text{new}}^{(t)} \mathbf{G}_2 =$

$$\left(\sum_{i \in V_{v_t}^{+1}} \mathbf{1}_i^{d_1'} - \sum_{i \in V_{v_t}^{+1} \setminus V_{\omega_t}} \mathbf{1}_{\ell+i}^{d_1'} + \sum_{i \in V_{\omega_t} \setminus V_{v_t}^{+1}} \mathbf{1}_{2\ell+1+i}^{d_1'} \right) \begin{array}{r|c|cccc|}
 & 0 & 1 & & & \ell \\
1 & & & & & \\
2 & & & 0 & & \\
\vdots & & & & & \\
\ell & & & & & \\
\hline
\ell+1 & & \mathbf{a} & & & \\
\ell+2 & 0 & & \mathbf{a} & & \\
 & & & & \ddots & \\
2\ell & & & & & \mathbf{a} \\
\hline
2\ell+1 & \mathbf{a} & & & & \\
2\ell+2 & & \mathbf{a} & & & \\
2\ell+3 & & & \mathbf{a} & & \\
 & & & & \ddots & \\
3\ell+1 & & & & & \mathbf{a} \\
\hline
 & & & 0 & & \\
\end{array}$$

$$= (\rho^{\langle t,0 \rangle} \mathbf{a}, \ldots, \rho^{\langle t,\ell \rangle} \mathbf{a}).$$

where we recall the definition of $\rho^{\langle t,i \rangle}$ from Eq. (39). Therefore, via EncBS' and EncR', the considering polynomial is substituted and evaluated to

$$\begin{aligned}
&\phi_p^{(t)} \mathbf{r}_{\text{new}}^{(t)} \mathbf{G}_2 + \sum_{\mu \in [m_{2,t}]} \phi_{p,\mu}^{(t)} \hat{\mathbf{r}}_\mu^{(2,t)} + \sum_{v \in [m_{1,t}], j \in [n]} \phi_{p,v,j}^{(t)} \mathbf{r}_v^{(2,t)} \mathbf{B}_{2,j} \\
&= \phi_p^{(t)} (\rho^{\langle t,0 \rangle} \mathbf{a}, \ldots, \rho^{\langle t,\ell \rangle} \mathbf{a}) + \sum_{\mu \in [m_{2,t}]} \phi_{p,\mu}^{(t)} (\rho^{\langle t,0 \rangle} \hat{\mathbf{r}}_\mu^{\langle t,0 \rangle}, \ldots, \rho^{\langle t,\ell \rangle} \hat{\mathbf{r}}_\mu^{\langle t,\ell \rangle}) \\
&\qquad + \sum_{v \in [m_{1,t}], j \in [n]} \phi_{p,v,j}^{(t)} (\rho^{\langle t,0 \rangle} \mathbf{r}_v^{\langle t,0 \rangle} \mathbf{B}_j^{(0)}, \ldots, \rho^{\langle t,\ell \rangle} \mathbf{r}_v^{\langle t,\ell \rangle} \mathbf{B}_j^{(\ell)}) \qquad (43) \\
&= \Bigg(\rho^{\langle t,0 \rangle} \Big(\phi_p^{(t)} \mathbf{a} + \sum_{\mu \in [m_{2,t}]} \phi_{p,\mu}^{(t)} \hat{\mathbf{r}}_\mu^{\langle t,0 \rangle} + \sum_{v \in [m_{1,t}], j \in [n]} \phi_{p,v,j}^{(t)} \mathbf{r}_v^{\langle t,0 \rangle} \mathbf{B}_j^{(0)} \Big), \ldots, \\
&\qquad \rho^{\langle t,\ell \rangle} \Big(\phi_p^{(t)} \mathbf{a} + \sum_{\mu \in [m_{2,t}]} \phi_{p,\mu}^{(t)} \hat{\mathbf{r}}_\mu^{\langle t,\ell \rangle} + \sum_{v \in [m_{1,t}], j \in [n]} \phi_{p,v,j}^{(t)} \mathbf{r}_v^{\langle t,\ell \rangle} \mathbf{B}_j^{(\ell)} \Big) \Bigg) \\
&\qquad (44)
\end{aligned}$$

where Eq. (43) holds since

$$\begin{aligned}\mathbf{r}_v^{(2,t)}\mathbf{B}_{2,j} &= (\rho^{\langle t,0\rangle}\mathbf{r}_v^{\langle t,0\rangle},\ldots,\rho^{\langle t,\ell\rangle}\mathbf{r}_v^{\langle t,\ell\rangle})\begin{pmatrix}\mathbf{B}_j^{(0)} & & & \\ & \mathbf{B}_j^{(1)} & & \\ & & \ddots & \\ & & & \mathbf{B}_j^{(\ell)}\end{pmatrix} \\ &= (\rho^{\langle t,0\rangle}\mathbf{r}_v^{\langle t,0\rangle}\mathbf{B}_j^{(0)},\ldots,\rho^{\langle t,\ell\rangle}\mathbf{r}_v^{\langle t,\ell\rangle}\mathbf{B}_j^{(\ell)}).\end{aligned}$$

We now consider the evaluation result, namely, the term (44), which is a vector in $\mathbb{Z}_N^{1\times d_1'}$. This can be divided to $\ell+1$ blocks each of length d_1. For $i\in[0,\ell]$, the block i is evaluated to exactly 0 since

- if $i\notin(V_{\upsilon_t}^{+1}\setminus V_{\omega_t})\cup(V_{\omega_t}\setminus V_{\upsilon_t}^{+1})$, then we have $\rho^{\langle t,0\rangle}=0$, by definition of $\rho^{\langle t,0\rangle}$.
- if $i\in(V_{\upsilon_t}^{+1}\setminus V_{\omega_t})\cup(V_{\omega_t}\setminus V_{\upsilon_t}^{+1})$, then we have $P_\kappa(x_t,y_i)=0$, due to Proposition 5. Hence, the polynomial in the block i evaluates to 0 due to the selective symbolic property of Γ.

Remark 6. In the above, we assume w.l.o.g. that $d_1'\geq 3\ell+1$, so that the non-zero rows of $\mathbf{H}_2$ (which comprise $3\ell+1$ rows) fits into the d_1' rows. This is w.l.o.g. since in case of $d_1'<3\ell+1$, we can just append all-zero row vectors to $\mathbf{B}_{2,j}$ (and all the d_1'-length vectors) to have $3\ell+1$ rows.

Co-selective Symbolic Property. We prove co-selective symbolic property of Γ' from that of Γ.

– $\mathsf{EncBR}'(M)$. First, parse $M=(Q,\mathcal{T},q_0,q_{\sigma-1})$. Further parse $Q=\{q_0,\ldots,q_{\sigma-1}\}$, and $\mathcal{T}=\left\{\ (q_{\upsilon_t},q_{\omega_t},x_t)\ \right\}_{t\in[m]}$. Proceed as follows.

1. For $t\in[m]$, run

$$\mathsf{EncBR}(x_t)\to\left(\mathbf{B}_1^{(t)},\ldots,\mathbf{B}_n^{(t)};\ \mathbf{r}_1^{(t)},\ldots,\mathbf{r}_{m_{1,t}}^{(t)};\ \mathbf{a},\hat{\mathbf{r}}_1^{(t)},\ldots,\hat{\mathbf{r}}_{m_{2,t}}^{(t)}\right).$$

where $\mathbf{B}_j^{(t)}\in\mathbb{Z}_N^{d_1\times d_2}$, $\mathbf{r}_v^{(t)}\in\mathbb{Z}_N^{1\times d_1}$, $\hat{\mathbf{r}}_\mu^{(t)}\in\mathbb{Z}_N^{1\times d_2}$, and $\mathbf{a}=\mathbf{1}_1^{d_2}\in\mathbb{Z}_N^{1\times d_2}$. Let $d_1'=md_1$ and $d_2'=2md_2$.

2. For $j \in [n]$, let

$$\mathbf{B}_{1,j} = \begin{array}{c} \begin{array}{cccc|ccc} 1 & 2 & \cdots & m & m+1 & \cdots & 2m \end{array} \\ \left(\begin{array}{cccc|c} \mathbf{B}_j^{(1)} & & & & \\ & \mathbf{B}_j^{(2)} & & & 0 \\ & & \ddots & & \\ & & & \mathbf{B}_j^{(m)} & \end{array}\right) \end{array} \in \mathbb{Z}_N^{d_1' \times d_2'},$$

$$\mathbf{B}_{2,j} = \begin{array}{c} \begin{array}{ccc|cccc} 1 & \cdots & m & m+1 & m+2 & \cdots & 2m \end{array} \\ \left(\begin{array}{c|cccc} & \mathbf{B}_j^{(1)} & & & \\ 0 & & \mathbf{B}_j^{(2)} & & \\ & & & \ddots & \\ & & & & \mathbf{B}_j^{(m)} \end{array}\right) \end{array} \in \mathbb{Z}_N^{d_1' \times d_2'}.$$

3. Let

$$\mathbf{G}_1 = \begin{array}{c} \begin{array}{cccc|ccc} 1 & 2 & \cdots & m & m+1 & \cdots & 2m \end{array} \\ \left(\begin{array}{cccc|c} \mathbf{a} & & & & \\ & \mathbf{a} & & & 0 \\ & & \ddots & & \\ & & & \mathbf{a} & \\ \hline \multicolumn{5}{c}{0} \end{array}\right) \end{array} \in \mathbb{Z}_N^{d_1' \times d_2'},$$

$$\mathbf{G}_2 = -\begin{array}{c} \begin{array}{ccc|cccc} 1 & \cdots & m & m+1 & m+2 & \cdots & 2m \end{array} \\ \left(\begin{array}{c|cccc} & \mathbf{a} & & & \\ 0 & & \mathbf{a} & & \\ & & & \ddots & \\ & & & & \mathbf{a} \\ \hline \multicolumn{5}{c}{0} \end{array}\right) \end{array} \in \mathbb{Z}_N^{d_1' \times d_2'}.$$

4. Let

$$\begin{aligned} \mathbf{H}_0 &= \mathbf{1}_{1,1}^{d_1' \times d_2'} \in \mathbb{Z}_N^{d_1' \times d_2'}, \\ \mathbf{H}_1 &= -\sum_{t \in [m]} \mathbf{1}_{t,(\upsilon_t d_2 + 1)}^{d_1' \times d_2'} \in \mathbb{Z}_N^{d_1' \times d_2'}, \\ \mathbf{H}_2 &= \sum_{t \in [m]} \mathbf{1}_{t,(\omega_t d_2 + 1)}^{d_1' \times d_2'} \in \mathbb{Z}_N^{d_1' \times d_2'}. \end{aligned}$$

5. For $t \in [m]$, $v \in [m_{1,t}]$, $\mu \in [m_{2,t}]$, let

$$\begin{aligned}
\mathbf{r}_v^{(1,t)} &= (0, \ldots, 0, \overset{\substack{\text{block}\, t \\ \downarrow}}{\mathbf{r}_v^{(t)}}, 0, \ldots, 0) && \in \mathbb{Z}_N^{1 \times d_1'}, \\
\hat{\mathbf{r}}_\mu^{(1,t)} &= (0, \ldots, 0, \overset{\substack{\text{block}\, t \\ \downarrow}}{\hat{\mathbf{r}}_\mu^{(t)}}, 0, \ldots, 0) && \in \mathbb{Z}_N^{1 \times d_2'} \\
\mathbf{r}_v^{(2,t)} &= -(0, \ldots, 0, \overset{\substack{\text{block}\, t \\ \downarrow}}{\mathbf{r}_v^{(t)}}, 0, \ldots, 0) && \in \mathbb{Z}_N^{1 \times d_1'}, \\
\hat{\mathbf{r}}_\mu^{(2,t)} &= -(0, \ldots, 0, \overset{\substack{\text{block}\, m+t \\ \downarrow}}{\hat{\mathbf{r}}_\mu^{(t)}}, 0, \ldots, 0) && \in \mathbb{Z}_N^{1 \times d_2'}
\end{aligned}$$

   where the vector of length $d_1' = md_1$ is divided to m blocks of length d_1, starting from block 1 to block m.[24] Blocks in d_2'-length vectors are defined similarly, but run through block $2m$.

6. Let $\mathbf{r}_{\text{new}}^{(0)} = \mathbf{1}_1^{d_1'}$. For $t \in [m]$, let

$$\mathbf{r}_{\text{new}}^{(t)} = \mathbf{1}_t^{d_1'} \in \mathbb{Z}_N^{1 \times d_1'}.$$

7. For $k \in [0, \sigma - 1]$, let

$$\mathbf{u}_k = \mathbf{1}_{kd_2+1}^{d_2'} \in \mathbb{Z}_N^{1 \times d_2'}.$$

8. Finally, output

$$\begin{aligned}
\mathbb{B}' &= \big(\mathbf{B}_{1,1}, \ldots, \mathbf{B}_{1,n}, \mathbf{B}_{2,1}, \ldots, \mathbf{B}_{2,n}, \mathbf{H}_0, \mathbf{G}_1, \mathbf{H}_1, \mathbf{G}_2, \mathbf{H}_2\big), \\
\mathbb{R}' &= \left(\mathbf{r}_{\text{new}}^{(0)}, \ldots, \mathbf{r}_{\text{new}}^{(m)}, \left(\mathbf{r}_1^{(1,t)}, \ldots, \mathbf{r}_{m_{1,t}}^{(1,t)}, \mathbf{r}_1^{(2,t)}, \ldots, \mathbf{r}_{m_{1,t}}^{(2,t)}\right)_{t \in [m]}\right), \\
\hat{\mathbb{R}}' &= \left(\mathbf{u}_{\sigma-1}, \mathbf{u}_0, \ldots, \mathbf{u}_{\sigma-2}, \left(\hat{\mathbf{r}}_1^{(1,t)}, \ldots, \hat{\mathbf{r}}_{m_{2,t}}^{(1,t)}, \hat{\mathbf{r}}_1^{(2,t)}, \ldots, \hat{\mathbf{r}}_{m_{2,t}}^{(2,t)}\right)_{t \in [m]}\right)
\end{aligned}$$

   which define substitutions $\mathbf{b}' : \mathbb{B}'$, $\mathbf{r}' : \mathbb{R}'$ and $\hat{\mathbf{r}}' : \mathbb{R}'$.

- $\mathsf{EncS}'(M, Y)$. Parse $Y = (y_1, \ldots, y_\ell)$. Parse $M = (Q, \mathcal{T}, q_0, q_{\sigma-1})$. Further parse $Q = \{q_0, \ldots, q_{\sigma-1}\}$, and $\mathcal{T} = \big\{\ (q_{v_t}, q_{\omega_t}, x_t)\ \big\}_{t \in [m]}$.

  1. For $t \in [m], i \in [\ell]$ such that $P_\kappa(x_t, y_i) = 0$, it is possible to run

$$\mathsf{EncS}(x_t, y_i) \to \left(\mathbf{s}_0^{\langle t,i \rangle}, \ldots, \mathbf{s}_{w_{1,i}}^{\langle t,i \rangle};\ \hat{\mathbf{s}}_1^{\langle t,i \rangle}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{\langle t,i \rangle}\right),$$

     where $\mathbf{s}_\tau^{\langle t,i \rangle} \in \mathbb{Z}_N^{1 \times d_2}$, $\hat{\mathbf{s}}_z^{\langle t,i \rangle} \in \mathbb{Z}_N^{1 \times d_1}$. Compute $\delta^{\langle t,i \rangle} := \big(1/\mathbf{s}_0^{\langle t,i \rangle}[1]\big)$, which is computable since $\mathbf{s}_0^{\langle t,i \rangle}[1] = \mathbf{a}(\mathbf{s}_0^{\langle t,i \rangle})^\top \neq 0$ (from the symbolic property of Γ).

[24] Hence, here, the i-th block contains elements of position $(i-1)d_1 + 1$ to id_1.

2. For $t \in [m], i \in [\ell]$, define[25]

$$\varphi^{\langle t,i\rangle} := \begin{cases} \delta^{\langle t,i\rangle} & \text{if } \upsilon_t \in U_{i-1} \text{ and } P_\kappa(x_t, y_i) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

$$\theta^{\langle t,i\rangle} := \begin{cases} \delta^{\langle t,i\rangle} & \text{if } \omega_t \in U_i \text{ and } P_\kappa(x_t, y_i) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

3. For $i \in [\ell]$, $\tau \in [w_{1,i}]^+$, first define some intermediate notations as follows.

$$\mathbf{e}_\tau^{(i)} := \left(\varphi^{\langle 1,i\rangle}\mathbf{s}_\tau^{\langle 1,i\rangle}, \ldots, \varphi^{\langle m,i\rangle}\mathbf{s}_\tau^{\langle m,i\rangle}\right) \in \mathbb{Z}_N^{1\times md_2}, \tag{47}$$

$$\mathbf{f}_\tau^{(i)} := \left(\theta^{\langle 1,i\rangle}\mathbf{s}_\tau^{\langle 1,i\rangle}, \ldots, \theta^{\langle m,i\rangle}\mathbf{s}_\tau^{\langle m,i\rangle}\right) \in \mathbb{Z}_N^{1\times md_2}. \tag{48}$$

   We also let $\mathbf{f}_\tau^{(0)} := 0$ and $\mathbf{e}_\tau^{(\ell+1)} := 0$. We then let

$$\mathbf{s}_\tau^{(1,i)} = (\mathbf{e}_\tau^{(i)} \quad, \mathbf{f}_\tau^{(i-1)}) \in \mathbb{Z}_N^{1\times d_2'},$$
$$\mathbf{s}_\tau^{(2,i)} = (\mathbf{e}_\tau^{(i+1)}, \mathbf{f}_\tau^{(i)} \quad) \in \mathbb{Z}_N^{1\times d_2'}.$$

   We can see that this is consistent with the requirement from Eq. (18), that is, $\mathbf{s}_0^{(1,i+1)} = \mathbf{s}_0^{(2,i)}$.

4. For $i \in [\ell]$, $z \in [w_{2,i}]$, let

$$\hat{\mathbf{s}}_z^{(1,i)} := \left(\varphi^{\langle 1,i\rangle}\hat{\mathbf{s}}_z^{\langle 1,i\rangle}, \ldots, \varphi^{\langle m,i\rangle}\hat{\mathbf{s}}_z^{\langle m,i\rangle}\right) \in \mathbb{Z}_N^{1\times d_1'},$$
$$\hat{\mathbf{s}}_z^{(2,i)} := \left(\theta^{\langle 1,i\rangle}\hat{\mathbf{s}}_z^{\langle 1,i\rangle}, \ldots, \theta^{\langle m,i\rangle}\hat{\mathbf{s}}_z^{\langle m,i\rangle}\right) \in \mathbb{Z}_N^{1\times d_1'}.$$

5. For $i \in [0, \ell]$, let $\mathbf{s}_{\text{new}}^{(i)} = (\mathbf{s}_{\text{new},0}^{(i)}, \ldots, \mathbf{s}_{\text{new},\sigma-1}^{(i)}, 0, \ldots, 0) \in \mathbb{Z}_N^{1\times d_2'}$, where for $k \in [0, \sigma - 1]$, each block $\mathbf{s}_{\text{new},k}^{(i)}$ is of length d_2 and

$$\mathbf{s}_{\text{new},k}^{(i)} := \begin{cases} \mathbf{1}_1^{d_2} & \text{if } k \in U_i \\ 0 & \text{if } k \notin U_i \end{cases}.$$

   That is, $\mathbf{s}_{\text{new}}^{(i)} = \sum_{k\in U_i} \mathbf{1}_{kd_2+1}^{d_2'}$.

6. Finally, output

$$\mathbb{S}' = \left(\mathbf{s}_{\text{new}}^{(\ell)}, \mathbf{s}_{\text{new}}^{(0)}, \ldots, \mathbf{s}_{\text{new}}^{(\ell-1)}, \left(\mathbf{s}_0^{(1,i)}, \ldots, \mathbf{s}_{w_{1,i}}^{(1,i)}, \mathbf{s}_0^{(2,i)}, \ldots, \mathbf{s}_{w_{1,i}}^{(2,i)}\right)_{i\in[\ell]}\right),$$
$$\hat{\mathbb{S}}' = \left(\hat{\mathbf{s}}_1^{(1,i)}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{(1,i)}, \hat{\mathbf{s}}_1^{(2,i)}, \ldots, \hat{\mathbf{s}}_{w_{2,i}}^{(2,i)}\right)_{i\in[\ell]}$$

   which define substitutions $\mathbf{s}' : \mathbb{S}'$, and $\hat{\mathbf{s}}' : \hat{\mathbb{S}}'$.

[25] Intuitively, $(\varphi^{\langle t,0\rangle}, \ldots, \varphi^{\langle t,\ell\rangle})$, $(\theta^{\langle t,0\rangle}, \ldots, \theta^{\langle t,\ell\rangle})$ will play the role of the "mask" vectors, as motivated at the end of §2: it encodes exactly the non-acceptance condition of DFA M (Proposition 5).

Verifying Properties. Since $U_\ell = \{\sigma - 1\}$ (*cf.* Proposition 4), we have $\mathbf{s}_{\mathrm{new}}^{(\ell)} = \mathbf{1}_{(\sigma-1)d_2+1}^{d_2'}$. We can thus verify that,

$$\mathbf{u}_{\sigma-1}(\mathbf{s}_{\mathrm{new}}^{(\ell)})^\top = \mathbf{1}_{(\sigma-1)d_2+1}^{d_2'}(\mathbf{1}_{(\sigma-1)d_2+1}^{d_2'})^\top = 1$$

which is not zero, as required.

For $\mathsf{key\text{-}enc}$, we have the following polynomials.

- The polynomial $\tilde{k}_0 = -u_0 + r_{\mathrm{new}}^{(0)} h_0$ is substituted and evaluated to

$$-\mathbf{u}_0 + \mathbf{r}_{\mathrm{new}}^{(0)}\mathbf{H}_0 = -\mathbf{1}_1^{d_2'} + \mathbf{1}_1^{d_1'} \cdot \mathbf{1}_{1,1}^{d_1' \times d_2'} = 0$$

- For $t \in [m]$, the polynomial $\tilde{k}_{1,t} = u_{\upsilon_t} + r_{\mathrm{new}}^{(t)} h_1$ is substituted and evaluated to

$$\mathbf{u}_{\upsilon_t} + \mathbf{r}_{\mathrm{new}}^{(t)}\mathbf{H}_1 = \mathbf{1}_{\upsilon_t d_2+1}^{d_2'} + \mathbf{1}_t^{d_1'} \cdot (-\sum_{t\in[m]} \mathbf{1}_{t,(\upsilon_t d_2+1)}^{d_1' \times d_2'}) = 0.$$

- For $t \in [m]$, the polynomial $\tilde{k}_{2,t} = -u_{\omega_t} + r_{\mathrm{new}}^{(t)} h_2$ is substituted and evaluated to

$$-\mathbf{u}_{\omega_t} + \mathbf{r}_{\mathrm{new}}^{(t)}\mathbf{H}_2 = -\mathbf{1}_{\omega_t d_2+1}^{d_2'} + \mathbf{1}_t^{d_1'} \cdot \sum_{t\in[m]} \mathbf{1}_{t,(\omega_t d_2+1)}^{d_1' \times d_2'} = 0.$$

- For $t \in [m]$, $p \in [m_{3,t}]$, the p-th polynomial in $\mathbf{k'}^{(1,t)}$ is

$$\phi_p^{(t)} r_{\mathrm{new}}^{(t)} g_1 + \sum_{\mu\in[m_{2,t}]} \phi_{p,\mu}^{(t)} \hat{r}_\mu^{(1,t)} + \sum_{v\in[m_{1,t}],j\in[n]} \phi_{p,v,j}^{(t)} r_v^{(1,t)} b_{1,j}.$$

where we recall that the coefficients are those of $\mathbf{k}^{(t)}$ obtained from $\mathsf{EncKey}(x_t, N)$ (for the PES $\mathit{\Gamma}$), and we replace $\alpha^{(1,t)}$ with $r_{\mathrm{new}}^{(t)} g_1$. Also note that $m_{3,t}$ is the size of $\mathbf{k}^{(t)}$. Via EncBR', it is substituted to

$$\phi_p^{(t)} \mathbf{r}_{\mathrm{new}}^{(t)} \mathbf{G}_1 + \sum_{\mu\in[m_{2,t}]} \phi_{p,\mu}^{(t)} \hat{\mathbf{r}}_\mu^{(1,t)} + \sum_{v\in[m_{1,t}],j\in[n]} \phi_{p,v,j}^{(t)} \mathbf{r}_v^{(1,t)} \mathbf{B}_{1,j}$$

which is evaluated to

$$\begin{aligned} &\phi_p^{(t)} \quad (0,\ldots,0,\ \overset{\substack{\text{block}\,t\\ \downarrow}}{\mathbf{a}}\ ,\quad 0,\ldots,0)+ \\ \sum_{u\in[m_{2,t}]} &\phi_{p,u}^{(t)} \ (0,\ldots,0,\ \overset{\substack{\text{block}\,t\\ \downarrow}}{\hat{\mathbf{r}}_u^{(t)}}\ ,\quad 0,\ldots,0)+ \\ \sum_{v\in[m_{1,t}],j\in[n]} &\phi_{p,v,j}^{(t)} (0,\ldots,0,\ \overset{\substack{\text{block}\,t\\ \downarrow}}{\mathbf{r}_v^{(t)}\mathbf{B}_j^{(t)}}\ ,0,\ldots,0) \end{aligned}$$

which is exactly 0 since the sum at the t-th block is 0 due to the co-selective symbolic property of $\mathit{\Gamma}$. Note that, in the above, we use

$$\mathbf{r}^{(t)}_{\text{new}}\mathbf{G}_1 = (0,\dots,0,\overset{\overset{t}{\downarrow}}{1},0,\dots,0)\cdot \begin{array}{c} \begin{array}{ccccccc}1 & 2 & \cdots & m & m{+}1 & \cdots & 2m\end{array} \\ \left(\begin{array}{cccc|c} \mathbf{a} & & & & \\ & \mathbf{a} & & & 0 \\ & & \ddots & & \\ & & & \mathbf{a} & \\ \hline \multicolumn{5}{c}{0} \end{array}\right)\end{array}$$

$$= (0,\dots,0,\ \overset{\overset{\text{block}\,t}{\downarrow}}{\mathbf{a}}\ ,0,\dots,0),$$

and

$$\mathbf{r}^{(1,t)}_v\mathbf{B}_{1,j} = (0,\dots,0,\ \overset{\overset{\text{block}\,t}{\downarrow}}{\mathbf{r}^{(t)}_v}\ ,0,\dots,0)\cdot \begin{array}{c} \begin{array}{ccccccc}1 & 2 & \cdots & m & m{+}1 & \cdots & 2m\end{array} \\ \left(\begin{array}{cccc|c} \mathbf{B}^{(1)}_j & & & & \\ & \mathbf{B}^{(2)}_j & & & 0 \\ & & \ddots & & \\ & & & \mathbf{B}^{(m)}_j & \end{array}\right)\end{array}$$

$$= (0,\dots,0,\ \overset{\overset{\text{block}\,t}{\downarrow}}{\mathbf{r}^{(t)}_v\mathbf{B}^{(t)}_j}\ ,0,\dots,0).$$

- For $t \in [m]$, $p \in [m_{3,t}]$, the p-th polynomial in $\mathbf{k'}^{(2,t)}$ is

$$\phi^{(t)}_p r^{(t)}_{\text{new}} g_2 + \sum_{\mu\in[m_{2,t}]} \phi^{(t)}_{p,\mu}\hat{r}^{(2,t)}_\mu + \sum_{v\in[m_{1,t}],j\in[n]} \phi^{(t)}_{p,v,j} r^{(2,t)}_v b_{2,j}.$$

  Via EncBR', it is substituted to

$$\phi^{(t)}_p \mathbf{r}^{(t)}_{\text{new}}\mathbf{G}_2 + \sum_{\mu\in[m_{2,t}]} \phi^{(t)}_{p,\mu}\hat{\mathbf{r}}^{(2,t)}_\mu + \sum_{v\in[m_{1,t}],j\in[n]} \phi^{(t)}_{p,v,j}\mathbf{r}^{(2,t)}_v\mathbf{B}_{2,j}$$

  which is evaluated to

$$\begin{aligned} &- \phi^{(t)}_p\ \ (0,\dots,0,\ \ \overset{\overset{\text{block}\,m+t}{\downarrow}}{\mathbf{a}}\ \ ,0,\dots,0)+ \\ \sum_{u\in[m_{2,t}]} &- \phi^{(t)}_{p,u}\ (0,\dots,0,\ \ \overset{\overset{\text{block}\,m+t}{\downarrow}}{\hat{\mathbf{r}}^{(t)}_u}\ \ ,0,\dots,0)+ \\ \sum_{v\in[m_{1,t}],j\in[n]} &- \phi^{(t)}_{p,v,j}(0,\dots,0,\ \overset{\overset{\text{block}\,m+t}{\downarrow}}{\mathbf{r}^{(t)}_v\mathbf{B}^{(t)}_j}\ ,0,\dots,0) \end{aligned}$$

which is exactly 0 since the sum at the $(m+t)$-th block is 0 due to the co-selective symbolic property of Γ. Note that, in the above, we use

$$\mathbf{r}^{(t)}_{\mathrm{new}}\mathbf{G}_2 = -(0,\ldots,0,\overset{\overset{t}{\downarrow}}{1},0,\ldots,0)\cdot\left(\begin{array}{ccc|cccc} 1 & \cdots & m & m{+}1 & m{+}2 & \cdots & 2m \\ & & & \mathbf{a} & & & \\ & 0 & & & \mathbf{a} & & \\ & & & & & \ddots & \\ & & & & & & \mathbf{a} \\ \hline & & & 0 & & & \end{array}\right)$$

$$= -(0,\ldots,0,\overset{\overset{\text{block } m+t}{\downarrow}}{\mathbf{a}},0,\ldots,0),$$

and

$$\mathbf{r}^{(2,t)}_{v}\mathbf{B}_{2,j} = -(0,\ldots,0,\overset{\overset{\text{block } t}{\downarrow}}{\mathbf{r}^{(t)}_{v}},0,\ldots,0)\cdot\left(\begin{array}{ccc|cccc} 1 & \cdots & m & m{+}1 & m{+}2 & \cdots & 2m \\ & & & \mathbf{B}^{(1)}_j & & & \\ & 0 & & & \mathbf{B}^{(2)}_j & & \\ & & & & & \ddots & \\ & & & & & & \mathbf{B}^{(m)}_j \end{array}\right)$$

$$= (0,\ldots,0,\overset{\overset{\text{block } m+t}{\downarrow}}{\mathbf{r}^{(t)}_{v}\mathbf{B}^{(t)}_j},0,\ldots,0).$$

For ct-enc, we have the following polynomials.

- The polynomial $c'_0 = h_0 s^{(0)}_{\mathrm{new}}$ is substituted and evaluated to

$$\mathbf{H}_0(\mathbf{s}^{(0)}_{\mathrm{new}})^\top = \mathbf{1}^{d'_1\times d'_2}_{1,1}\cdot(\sum_{k\in U_0}\mathbf{1}^{d'_2}_{kd_2+1})^\top = 0,$$

  since $0 \notin U_0$ due to Proposition 5.
- For $i \in [\ell]$, the polynomial $c'_i = h_1 s^{(i-1)}_{\mathrm{new}} + g_1 s'^{(i-1)}_0 + h_2 s^{(i)}_{\mathrm{new}} + g_2 s'^{(i)}_0$ is substituted and evaluated as follows.
  - The first term, $h_1 s^{(i-1)}_{\mathrm{new}}$, is substituted and evaluated to

$$\mathbf{H}_1(\mathbf{s}^{(i-1)}_{\mathrm{new}})^\top = \left(-\sum_{t\in[m]}\mathbf{1}^{d'_1\times d'_2}_{t,(\upsilon_t d_2+1)}\right)\cdot\left(\sum_{k\in U_{i-1}}\mathbf{1}^{d'_2}_{kd_2+1}\right)^\top$$
$$= -\sum_{t \text{ s.t. } \upsilon_t\in U_{i-1}}(\mathbf{1}^{d'_1}_t)^\top.$$

- The third term, $h_2 s_{\text{new}}^{(i)}$, is substituted and evaluated to

$$\begin{aligned}\mathbf{H}_2(\mathbf{s}_{\text{new}}^{(i)})^\top &= \left(\sum_{t\in[m]} \mathbf{1}_{t,(\omega_t d_2+1)}^{d_1'\times d_2'}\right)\cdot\left(\sum_{k\in U_i}\mathbf{1}_{kd_2+1}^{d_2'}\right)^\top \\ &= \sum_{t \text{ s.t. } \omega_t\in U_i}(\mathbf{1}_t^{d_1'})^\top.\end{aligned}$$

- The second term, $g_1 s_0'^{(i-1)}$, is considered as follows. From Eq. (18), we have $g_1 s_0'^{(i-1)} = g_1 s_0^{(1,i)}$. It is then substituted and evaluated to

$$\mathbf{G}_1(\mathbf{s}_0^{(1,i)})^\top = \begin{array}{c} \begin{array}{ccccccc} {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle \cdots} & {\scriptstyle m} & {\scriptstyle m+1} & {\scriptstyle \cdots} & {\scriptstyle 2m} \end{array} \\ \left(\begin{array}{cccc|c} \mathbf{a} & & & & \\ & \mathbf{a} & & & 0 \\ & & \ddots & & \\ & & & \mathbf{a} & \\ \hline \multicolumn{5}{c}{0} \end{array}\right) \end{array} \cdot (\mathbf{e}_0^{(i)}, \mathbf{f}_0^{(i-1)})^\top$$

$$= \begin{array}{c} \begin{array}{cccc} {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle \cdots} & {\scriptstyle m} \end{array} \\ \left(\begin{array}{cccc} \mathbf{a} & & & \\ & \mathbf{a} & & \\ & & \ddots & \\ & & & \mathbf{a} \\ \hline \multicolumn{4}{c}{0} \end{array}\right) \end{array} \cdot (\mathbf{e}_0^{(i)})^\top \tag{49}$$

$$= \left(\varphi^{\langle 1,i\rangle}\mathbf{a}(\mathbf{s}_\tau^{\langle 1,i\rangle})^\top, \ldots, \varphi^{\langle m,i\rangle}\mathbf{a}(\mathbf{s}_\tau^{\langle m,i\rangle})^\top, 0, \ldots, 0\right)^\top \tag{50}$$

$$= \sum_{\substack{t \text{ s.t. } \upsilon_t\in U_{i-1}\wedge \\ P_\kappa(x_t,y_i)=0}} (\mathbf{1}_t^{d_1'})^\top \tag{51}$$

where Eq. (50) is due to the definition of $\mathbf{e}_0^{(i)}$ from Eq. (47), that is, $\mathbf{e}_0^{(i)} = \left(\varphi^{\langle 1,i\rangle}\mathbf{s}_\tau^{\langle 1,i\rangle}, \ldots, \varphi^{\langle m,i\rangle}\mathbf{s}_\tau^{\langle m,i\rangle}\right)$, while Eq. (51) is due to the definition of $\varphi^{\langle t,i\rangle}$ given in Eq. (45).

- The fourth term, $g_2 s_0'^{(i)}$, is considered as follows. From Eq. (18), we have $g_2 s_0'^{(i)} = g_2 s_0^{(2,i)}$. It is then substituted and evaluated to

$$\mathbf{G}_2(\mathbf{s}_0^{(2,i)})^\top = -\begin{array}{c|cccc} 1 \quad \cdots \quad m & m{+}1 & m{+}2 & \cdots & 2m \\ \left(\begin{array}{c} \\ \\ 0 \\ \\ \end{array}\right. & \mathbf{a} & & & \\ & & \mathbf{a} & & \\ & & & \ddots & \\ & & & & \left.\mathbf{a}\right) \\ \hline \multicolumn{5}{c}{0} \end{array} \cdot (\mathbf{e}_0^{(i+1)}, \mathbf{f}_0^{(i)})^\top$$

$$= -\begin{array}{cccc} 1 & 2 & \cdots & m \\ \left(\mathbf{a}\right. & & & \\ & \mathbf{a} & & \\ & & \ddots & \\ & & & \left.\mathbf{a}\right) \\ \hline \multicolumn{4}{c}{0} \end{array} \cdot (\mathbf{f}_0^{(i)})^\top \tag{52}$$

$$= -\Big(\theta^{\langle 1,i\rangle}\mathbf{a}(\mathbf{s}_\tau^{\langle 1,i\rangle})^\top, \ldots, \theta^{\langle m,i\rangle}\mathbf{a}(\mathbf{s}_\tau^{\langle m,i\rangle})^\top, 0, \ldots, 0\Big)^\top \tag{53}$$

$$= -\sum_{\substack{t \text{ s.t. } \omega_t \in U_i \wedge \\ P_\kappa(x_t, y_i)=0}} (\mathbf{1}_t^{d_1'})^\top \tag{54}$$

where Eq. (53) is due to the definition of $\mathbf{f}_0^{(i)}$ from Eq. (48), that is, $\mathbf{f}_0^{(i)} = \Big(\theta^{\langle 1,i\rangle}\mathbf{s}_\tau^{\langle 1,i\rangle}, \ldots, \theta^{\langle m,i\rangle}\mathbf{s}_\tau^{\langle m,i\rangle}\Big)$, while Eq. (54) is due to the definition of $\theta^{\langle t,i\rangle}$ given in Eq. (46).

- Combining all the four substituted terms, we have that the substitution for c_i' is evaluated to $\mathbf{H}_1(\mathbf{s}_{\text{new}}^{(i-1)})^\top + \mathbf{G}_1(\mathbf{s}_0^{(1,i)})^\top + \mathbf{H}_2(\mathbf{s}_{\text{new}}^{(i)})^\top + \mathbf{G}_2(\mathbf{s}_0^{(2,i)})^\top$

$$\begin{aligned} = \quad & -\sum_{t \text{ s.t. } \upsilon_t \in U_{i-1}} (\mathbf{1}_t^{d_1'})^\top + \sum_{\substack{t \text{ s.t. } \upsilon_t \in U_{i-1} \wedge \\ P_\kappa(x_t, y_i)=0}} (\mathbf{1}_t^{d_1'})^\top \\ & + \sum_{t \text{ s.t. } \omega_t \in U_i} (\mathbf{1}_t^{d_1'})^\top - \sum_{\substack{t \text{ s.t. } \omega_t \in U_i \wedge \\ P_\kappa(x_t, y_i)=0}} (\mathbf{1}_t^{d_1'})^\top \\ = \quad & -\sum_{\substack{t \text{ s.t. } \upsilon_t \in U_{i-1} \wedge \\ P_\kappa(x_t, y_i)=1}} (\mathbf{1}_t^{d_1'})^\top + \sum_{\substack{t \text{ s.t. } \omega_t \in U_i \wedge \\ P_\kappa(x_t, y_i)=1}} (\mathbf{1}_t^{d_1'})^\top = 0. \end{aligned}$$

The last equality holds due to Proposition 5, which states that when M does not accept Y, the condition that $P_\kappa(x_t, y_i) = 1$ implies that we have $\upsilon_t \in U_{i-1}$ if and only if $\omega_t \in U_i$.

– For $i \in [\ell]$, $p \in [w_{3,i}]$, the p-th polynomial in $\mathbf{c}^{(1,i)}$ is

$$\sum_{z\in[w_{2,i}]} \eta^{(i)}_{p,z}\hat{s}^{(1,i)}_z + \sum_{\tau\in[w_{1,i}]^+, j\in[n]} \eta^{(i)}_{p,t,j} b_{1,j} s^{(1,i)}_\tau.$$

Via EncBR' and EncS', it is substituted to

$$\sum_{z\in[w_{2,i}]} \eta^{(i)}_{p,z}(\hat{\mathbf{s}}^{(1,i)}_z)^\top + \sum_{\tau\in[w_{1,i}]^+, j\in[n]} \eta^{(i)}_{p,t,j}\mathbf{B}_{1,j}(\mathbf{s}^{(1,i)}_\tau)^\top$$

$$= \sum_{z\in[w_{2,i}]} \eta^{(i)}_{p,z}\begin{pmatrix} \varphi^{\langle 1,i\rangle}(\hat{\mathbf{s}}^{\langle 1,i\rangle}_z)^\top \\ \vdots \\ \varphi^{\langle m,i\rangle}(\hat{\mathbf{s}}^{\langle m,i\rangle}_z)^\top \end{pmatrix} + \sum_{\tau\in[w_{1,i}]^+, j\in[n]} \eta^{(i)}_{p,t,j}\begin{pmatrix} \varphi^{\langle 1,i\rangle}\mathbf{B}^{(1)}_j(\mathbf{s}^{\langle 1,i\rangle}_\tau)^\top \\ \vdots \\ \varphi^{\langle m,i\rangle}\mathbf{B}^{(m)}_j(\mathbf{s}^{\langle m,i\rangle}_\tau)^\top \end{pmatrix} \tag{55}$$

$$= \begin{pmatrix} \varphi^{\langle 1,i\rangle}(\sum_{z\in[w_{2,i}]}\eta^{(i)}_{p,z}(\hat{\mathbf{s}}^{\langle 1,i\rangle}_z)^\top + \sum_{\tau\in[w_{1,i}]^+, j\in[n]}\eta^{(i)}_{p,t,j}(\mathbf{s}^{\langle 1,i\rangle}_\tau)^\top) \\ \vdots \\ \varphi^{\langle m,i\rangle}(\sum_{z\in[w_{2,i}]}\eta^{(i)}_{p,z}(\hat{\mathbf{s}}^{\langle m,i\rangle}_z)^\top + \sum_{\tau\in[w_{1,i}]^+, j\in[n]}\eta^{(i)}_{p,t,j}(\mathbf{s}^{\langle m,i\rangle}_\tau)^\top) \end{pmatrix}. \tag{56}$$

Note that in Eq. (55) above, we use

$$\mathbf{B}_{1,j}(\mathbf{s}^{(1,i)}_\tau)^\top = \left(\begin{array}{cccc|ccc} 1 & 2 & \cdots & m & m{+}1 & \cdots & 2m \\ \mathbf{B}^{(1)}_j & & & & & & \\ & \mathbf{B}^{(2)}_j & & & & 0 & \\ & & \ddots & & & & \\ & & & \mathbf{B}^{(m)}_j & & & \end{array}\right) \cdot (\mathbf{e}^{(i)}_\tau, \mathbf{f}^{(i-1)}_\tau)^\top \tag{57}$$

$$= \left(\begin{array}{cccc} 1 & 2 & \cdots & m \\ \mathbf{B}^{(1)}_j & & & \\ & \mathbf{B}^{(2)}_j & & \\ & & \ddots & \\ & & & \mathbf{B}^{(m)}_j \end{array}\right) \cdot \begin{pmatrix} \varphi^{\langle 1,i\rangle}(\mathbf{s}^{\langle 1,i\rangle}_\tau)^\top \\ \vdots \\ \varphi^{\langle m,i\rangle}(\mathbf{s}^{\langle m,i\rangle}_\tau)^\top \end{pmatrix}$$

$$= \begin{pmatrix} \varphi^{\langle 1,i\rangle}\mathbf{B}^{(1)}_j(\mathbf{s}^{\langle 1,i\rangle}_\tau)^\top \\ \vdots \\ \varphi^{\langle m,i\rangle}\mathbf{B}^{(m)}_j(\mathbf{s}^{\langle m,i\rangle}_\tau)^\top \end{pmatrix}$$

The term (56) is a vector in $\mathbb{Z}_N^{d_1'\times 1}$. This can be divided to m blocks each of length d_1. For all $t \in [m]$, the t-th block is evaluated to exactly 0 since

- if t satisfies $v_t \in U_{i-1}$ and $P_\kappa(x_t, y_i) = 0$, then we have in particular that $P_\kappa(x_t, y_i) = 0$. Hence, the polynomial in the t-th block evaluates to 0 due to the co-selective symbolic property of $\mathit{\Gamma}$.

- if t does not satisfy the above, we have that $\varphi^{\langle t,i\rangle} = 0$ due to the definition of $\varphi^{\langle t,i\rangle}$.

– For $i \in [\ell]$, $p \in [w_{3,i}]$, the p-th polynomial in $\mathbf{c}^{(2,i)}$ is

$$\sum_{z\in[w_{2,i}]} \eta^{(i)}_{p,z}\hat{s}^{(2,i)}_z + \sum_{\tau\in[w_{1,i}]^+, j\in[n]} \eta^{(i)}_{p,t,j} b_{2,j} s^{(2,i)}_\tau .$$

Via EncBR' and EncS', it is substituted and evaluated to

$$\sum_{z\in[w_{2,i}]} \eta^{(i)}_{p,z}(\hat{\mathbf{s}}^{(2,i)}_z)^\top + \sum_{\tau\in[w_{1,i}]^+, j\in[n]} \eta^{(i)}_{p,t,j} \mathbf{B}_{2,j}(\mathbf{s}^{(2,i)}_\tau)^\top$$

which is evaluated to 0 in an analogous manner to the case of $\mathbf{c}^{(1,i)}$. Indeed, it is evaluated to exactly the term in Eq. (56) albeit replacing all $\varphi^{\langle t,i\rangle}$ with $\theta^{\langle t,i\rangle}$, and we use the analogous property of $\theta^{\langle t,i\rangle}$ to argue zero evaluation. Note that, in contrast to Eq. (57), this time we instead have

$$\mathbf{B}_{2,j}(\mathbf{s}^{(2,i)}_\tau)^\top = \begin{array}{ccc|cccc} 1 & \cdots & m & m{+}1 & m{+}2 & \cdots & 2m \\ & & & \mathbf{B}^{(1)}_j & & & \\ & 0 & & & \mathbf{B}^{(2)}_j & & \\ & & & & & \ddots & \\ & & & & & & \mathbf{B}^{(m)}_j \end{array} \cdot (\mathbf{e}^{(i+1)}_\tau, \mathbf{f}^{(i)}_\tau)^\top$$

$$= \begin{array}{cccc} 1 & 2 & \cdots & m \\ \mathbf{B}^{(1)}_j & & & \\ & \mathbf{B}^{(2)}_j & & \\ & & \ddots & \\ & & & \mathbf{B}^{(m)}_j \end{array} \cdot (\mathbf{f}^{(i)}_\tau)^\top .$$

J Proof for Predicative Branching Program

This section provides the implication from the predicate for predicative span programs to the predicate for predicative branching programs, which we omitted from 9.1. For formality, we first capture the definition as follows.

Definition 13. 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-Branching-augmented predicate* over P as $\mathsf{KB1}[P] = \{ \bar{P}_\kappa \}_\kappa$ where $\bar{P}_\kappa : \bar{\mathcal{X}}_\kappa \times \bar{\mathcal{Y}}_\kappa \to \{ 0,1 \}$ by letting

– $\bar{\mathcal{X}}_\kappa = \{ M \mid M$ is a predicative branching program over $P_\kappa \ \}$.
– $\bar{\mathcal{Y}}_\kappa = \mathcal{Y}_\kappa$.
– $\bar{P}_\kappa(M, y) = 1 \iff M$ accepts y. ◊

Augmentation over a set of predicates $\mathcal{P}$ can be defined analogously as previously, namely, we define $\mathsf{KB}[\mathcal{P}] := \mathsf{KB1}[\mathsf{CP1}_{\mathsf{OR}}[\mathsf{DS}[\mathcal{P}]]]$.

As a further note, we remark that, similarly to boolean branching programs in [23,8], it is w.l.o.g. to consider that there exists at most one edge connecting any two nodes (that is why we can let $E \subseteq V^2$), and that we have only one start node and one accept node.

We have the following result by generalizing the result that ABE for MSP implies ABE for branching program from [8].

Lemma 7. $\mathsf{KB1}[P]$ *can be embedded into* $\mathsf{KP1}[P]$.

The proof will follow in exactly the same manner as in [8]. We adapt it to the context of PBP and provide here for self-containment.

Proof. We define a map from a PBP $M = (\Gamma, q_1, q_\sigma, L)$ to an MSP $(\mathbf{A}, \pi)$ as follows. Parse the set of edges as $E = \{e_1, \ldots, e_m\}$. We represent an edge as $e_j = (q_{\upsilon_j}, q_{\omega_j})$, which means that the edge e_j directs from node q_{υ_j} to node q_{ω_j}. Let $\mathbf{A}$ be a matrix of dimension $m \times (\sigma - 1)$ where its entry at (j,k) is defined by

$$\mathbf{A}_{j,k} = \begin{cases} -1 & \text{if } k = \upsilon_j \\ 1 & \text{if } k = \omega_j \\ 0 & \text{otherwise} \end{cases} .$$

Let $\pi : [m] \to \mathcal{Y}_\kappa$ map $j \mapsto L(e_j)$. The map for ciphertext attribute is just the identity map. We now prove that M accepts y if and only if $(\mathbf{A}, \pi)$ accepts y.

We first prove the forward direction. Suppose M accepts y. Hence, there exists a path from the start node (node q_σ) to the accept node (node q_1) in Γ_y. Let $(e_{j_1}, \ldots, e_{j_t})$ be the edges on this path. Hence, $P_\kappa(L(e_{j_i}), y) = 1$ for all $i \in [1,t]$. Also, we have $\upsilon_{j_i} = \omega_{j_{1-1}}$ for all $i \in [2,t]$, while $\upsilon_{j_1} = \sigma$ and $\omega_{j_t} = 1$. Recall that $\mathbf{A}|_y$ consists of exactly all the rows j where $P_\kappa(\pi(j), y) = 1$; therefore, all the rows $j_1, \ldots, j_t$ are included in $\mathbf{A}|_y$, since $\pi(j_i) = L(e_{j_i})$. Now, consider the sum of rows $j_1, \ldots, j_t$. We have:

- The row j_1 contributes 1 at column ω_{j_1} (and, by the definition of $\mathbf{A}$, there is no column corresponding to $\upsilon_{j_1} = \sigma$).
- For $i \in [2, t-1]$, the row j_i contributes -1 at column $\upsilon_{j_i} = \omega_{j_{1-1}}$, and 1 at column ω_{j_i}.
- The row j_t contributes -1 at column $\upsilon_{j_t} = \omega_{j_{t-1}}$, and 1 at column $\omega_{j_t} = 1$.

Hence, all the values at column $\omega_{j_1}, \ldots, \omega_{j_{t-1}}$ are canceled out to 0, and it leaves only 1 at the column $\omega_{j_t} = 1$. That is, the sum is exactly $(1, 0, \ldots, 0)$. Therefore, $(1, 0, \ldots, 0) \in \mathrm{span}(\mathbf{A}|_y)$ which means that $(\mathbf{A}, \pi)$ accepts y.

We now prove the converse by contrapositive. Suppose that M does not accept y. Let Γ'_y be the *undirected* graph obtained from Γ_y by treating every edge as an undirected edge. We have the following properties:[26]

[26] These properties were also used, albeit differently, for proving selective security for the ABE for branching program of [23].

1. Since M does not accept y, we have that the start node and the accept node lie in different connected components of $\mathit{\Gamma}'_y$.
2. $\mathit{\Gamma}'_y$ contains no cycle. This is since $\mathit{\Gamma}_y$ is acyclic and every non-terminal node has exactly one outgoing edge due to the determinism of PBP.

Assume for the sake of contradiction that $(1, 0, \ldots, 0) \in \mathrm{span}(\mathbf{A}|_y)$. Let $J = \{ j \in [1, m] \mid P_\kappa(\pi(j), y) = 1 \}$. (Hence, J is the set of edge indexes of $\mathit{\Gamma}_y$). We write this linear combination as $(1, 0, \ldots, 0) = \sum_{j \in J} c_j \mathbf{A}_{j:}$, where $\mathbf{A}_{j:}$ is the row j of $\mathbf{A}$ and c_j is some coefficient. For each node index $k \in [1, \sigma]$, let J_k be the set of edge indexes j in J that are adjacent to k and that $c_j \neq 0$. From the linear combination, we must have that:

$$\sum_{j \in J_k} c_j \mathbf{A}_{j,k} = \begin{cases} 1 & \text{if } k = 1, \\ 0 & \text{if } k \in [2, \sigma - 1]. \end{cases} \tag{58}$$

Let $\mathit{\Gamma}''_y$ be the subgraph of $\mathit{\Gamma}'_y$ that takes all the edge indexes $j \in J$ such that $c_j \neq 0$. We observe that for every node index $k \in [2, \sigma - 1]$ (*i.e.,* not the accept nor the start node), there are *at least two edges adjacent to q_k in $\mathit{\Gamma}''_y$*. This is since otherwise the sum $\sum_{j \in J_k} c_j \mathbf{A}_{j,k}$ would not be canceled out to 0, where we observe that for $j \in J_k$ we have $\mathbf{A}_{j,k} \neq 0$. Next, we claim that $\mathit{\Gamma}''_y$ will always contain a cycle. Hence, this will contradict the property 2, and the proof will be concluded. It now remains to prove the claim. We consider an arbitrary node index $k \in [2, \sigma - 1]$. We have three cases:

– If q_k is connected to neither the accept nor the start node in $\mathit{\Gamma}''_y$, then in the largest connected subgraph of $\mathit{\Gamma}''_y$ that contains q_k, all the nodes have at least two edges adjacent to it.
– If q_k is connected to the accept node, then it is not connected to the start node by the property 1. Hence, in the largest connected subgraph of $\mathit{\Gamma}''_y$ that contains q_k, there exists exactly one node (*i.e.,* the accept node, q_1) that may have only one adjacent edge.
– If q_k is connected to the start node, then it is not connected to the accept node by the property 1. Hence, in the largest connected subgraph of $\mathit{\Gamma}''_y$ that contains q_k, there exists exactly one node (*i.e.,* the start node, q_σ) that has one adjacent edge.

In all three cases, the considering connected subgraph has at most one node that may have one adjacent edge (and all the other nodes have at least two adjacent edges). Hence, it always contain a cycle. This concludes the proof of the claim, and hence the lemma. □

K Unbounded Arbitrary Mixed-policy Augmentation

Nested-policy ABE, as proposed in §9, allows to nest ciphertext-policy and key-policy layers. However, the structure of nesting is fixed for the resulting augmented predicate. For example, $\mathsf{CP}[\mathsf{KP}(\mathcal{P})]$ is a nested predicate which has key-policy in a lower layer and ciphertext-policy in an upper layer. In this section, we will explore what we call "Mixed-policy" ABE where the nesting structure can be defined in an on-the-fly manner to a key and a ciphertext. In particular, the nesting structure is not fixed for the predicate definition (and hence is not fixed at the setup of the ABE scheme). The definition of such predicate will depend on a base set $\mathcal{P}$. The predicate allows any "closures" of $\mathcal{P}$ via operation KP and CP.

Definition 14. The *any-policy-augmented predicate* over a set $\mathcal{P}$ of predicate families set is defined as $\mathsf{KP}[\mathcal{P}] \odot \mathsf{CP}[\mathcal{P}]$. We will use $\mathsf{AP}[\mathcal{P}]$ as its shorthand.

Definition 15. The *ℓ-level mixed-policy-augmented predicate* over a set $\mathcal{P}$ of predicate families, denoted as $\mathsf{MP}_\ell[\mathcal{P}]$, is defined recursively as follows.

$$\begin{aligned} \mathsf{MP}_1[\mathcal{P}] &:= \mathsf{DS}[\mathcal{P}], \\ \mathsf{MP}_\ell[\mathcal{P}] &:= \mathsf{MP}_{\ell-1}[\mathcal{P}] \odot \mathsf{AP}[\mathsf{MP}_{\ell-1}[\mathcal{P}]]. \end{aligned}$$

$\diamondsuit$

This recursively defined predicate will contain arbitrarily augmented predicates so far to that level. To illustrate this, if we start from a set $\mathcal{P}$ of predicate families, at the second level we obtain predicate $\mathsf{DS}[\mathcal{P}] \odot \mathsf{KP}[\mathcal{P}] \odot \mathsf{CP}[\mathcal{P}]$. At the third level, we obtain predicate

$$\begin{aligned} \mathsf{DS}[\mathcal{P}] \odot \mathsf{KP}[\mathcal{P}] \odot \mathsf{CP}[\mathcal{P}] \odot \mathsf{KP}\Big[\mathsf{DS}[\mathcal{P}] \odot \mathsf{KP}[\mathcal{P}] \odot \mathsf{CP}[\mathcal{P}]\Big] \\ \odot \mathsf{CP}\Big[\mathsf{DS}[\mathcal{P}] \odot \mathsf{KP}[\mathcal{P}] \odot \mathsf{CP}[\mathcal{P}]\Big]. \end{aligned}$$

Hence, at the third level, any nested policy with two or less applications of $\mathsf{KP}[\cdot]$ and $\mathsf{CP}[\cdot]$, *e.g.,* $\mathsf{CP}[\mathsf{KP}(\mathcal{P})]$, will be contained as a special case.

Constructing PES for $\mathsf{MP}_\ell[\mathcal{P}]$. We first observe that using the concatenation direct sum construction, $\mathsf{Concat\text{-}Trans}$, would yield the parameter size being *exponential* in the number of levels ℓ. This is since when doing the direct sum with new schemes, the parameter sizes will be added up. When going from level $\ell-1$ to ℓ, the number of parameters for level ℓ will become at least 3 times of that for level $\ell-1$. Hence, the overall size at level ℓ would be $O(3^\ell)$.

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. Our construction is as follows. A PES for $\mathsf{MP}_1[\mathcal{P}]$ is obtained via $\mathsf{Reuse\text{-}Trans}$. From $\mathsf{MP}_{\ell-1}[\mathcal{P}]$, we obtain a PES for $\mathsf{KP}[\mathsf{MP}_{\ell-1}[\mathcal{P}]]$ via $\mathsf{CP1}_{\mathsf{OR}}$ and $\mathsf{KP1}$ as in Lemma 2, and a PES for $\mathsf{CP}[\mathsf{MP}_{\ell-1}[\mathcal{P}]]$ analogously. We then again use $\mathsf{Reuse\text{-}Trans}$ to combine to a PES for the direct sum of $\mathsf{MP}_{\ell-1}[\mathcal{P}]$, $\mathsf{KP}[\mathsf{MP}_{\ell-1}[\mathcal{P}]]$, and $\mathsf{CP}[\mathsf{MP}_{\ell-1}[\mathcal{P}]]$, which results in $\mathsf{MP}_\ell[\mathcal{P}]$. We have the following lemma.

Lemma 8. *Let $\mathcal{P} = \{P^{(1)}, \dots, P^{(k)}\}$ be a set of predicate families. Suppose that there exists a symbolically secure PES $\Gamma^{(j)}$ for $P^{(j)}$ each with the parameter size n_j. Let $n = \max_{j\in[k]} n_j$. Then, the above PES construction for $\mathsf{MP}_\ell[\mathcal{P}]$ has parameter size $n + 2k + 10\ell$.*

Proof. We prove by induction on ℓ. When $\ell = 1$, we have that the $\mathsf{Reuse\text{-}Trans}$ yield $n+2k$ parameters, as shown in Construction 5. Assume that $P' := \mathsf{MP}_{\ell-1}[\mathcal{P}]$ has parameter size $s := n+2k+10(\ell-1)$. Now since, $\mathsf{CP1\text{-}Trans}$ (for $\mathsf{CP1}_{\mathsf{OR}}[\cdot]$) and $\mathsf{KP1\text{-}Trans}$ (for $\mathsf{KP1}[\cdot]$) each adds up 2 elements, we have that a PES for $\mathsf{KP}[P']$ has parameter size $s + 4$. The same goes for $\mathsf{CP}[P']$. Note that for simplicity here, we always implicitly apply $\mathsf{Layer\text{-}Trans}$ so as to obtained admissible PES, which can then be used for $\mathsf{CP1\text{-}Trans}$, $\mathsf{KP1\text{-}Trans}$. That is why the additional 2 elements are counted. Then combining P', $\mathsf{KP}[P']$, $\mathsf{CP}[P']$ via $\mathsf{Reuse\text{-}Trans}$ yields a PES for $\mathsf{MP}_\ell[\mathcal{P}]$, which has parameter size $\max\{s, s+4, s+4\}+2(3) = s+10 = n + 2k + 10\ell$. □

L Further Discussions

This section gathers discussions that are deferred from various sections in the paper body.

L.1 On Difficulty of ABE for Non-monotone Span Programs

In §9, we describe how to achieve ABE for NSP in a modular manner. Here, we further discuss why ABE for NSP is seemingly more difficult to achieve than ABE for MSP.

In fact, there seems to some misunderstanding in the literature that ABE for MSP can be converted to ABE for NSP with only small (possibly constant) loss factor in efficiency (as we get comments from some anonymous reviewers). This seems to stem from the fact that a *Boolean* NSP can be interpreted as a *Boolean* MSP with the cost of increasing the input domain from, say n-bit, to $2n$-bit. This can be done by propagating all the internal NOT gates (think of it as a Boolean formula) so that they appear only at the input gates.

For the large-universe scheme, the input 0 and 1 amounts to check attribute membership and non-membership, say, $x \in Y$ and $x \notin Y$, respectively. Intuitively, in the context of ABE, this would be done by preparing the negative version of attributes in $\mathcal{U}$. However, if ABE for MSP is the only tool, this is not possible, since the functionality of checking if $x \notin Y$ would have to be done only via checking if $x \in \mathcal{U} \setminus Y$, which is, in turn, not clear how to achieve (in ABE) if $\mathcal{U}$ is of super-polynomial size. Moreover, even if $\mathcal{U}$ is of polynomial size, a ciphertext of any set Y would incur the fixed-once-and-for-all size of $O(|\mathcal{U}|)$, which can be inefficient. This issue is clearly discussed and motivated nicely in the original paper of Ostrovsky *et al.* [32], to which we would refer readers for further intuition.

L.2 Remark on the q-ratio Assumption

We remark that criticisms towards the so-called q-type assumptions often stem from the Cheon attack [18], which exactly exploits the property that an assumption contains terms like g^a and g^{a^q}, for some $q \in \mathbb{N}$, together in the same instance. However, there are no such terms in the q-ratio assumption of [2], where every variable contains only degree one or minus one (we refer its description to [2]). Hence this same reason for general criticism is not applied for the q-ratio assumption. (We do not argue that the q-ratio Assumption is all trustworthy, but argue only that the reason for criticism should not be from this general one by the Cheon attack).

We note further that the q-raio assumption has been shown in [2] to be implied from other specific q-type assumptions from [30,7], where the Cheon attack may apply (since they contain terms as above). However, this should not be a concern either since the q-raio assumption can be shown to hold in the generic group model in its own right, in a very simple manner. (We do not pursue here though.)

L.3 More Applications and Extensions

This section provides more applications, for supplementary to §9.

Extension for Direct Sum. Direct sum of Definition 7 can be generalized so that we can have many instances of one predicate family. To do so, an index is generalized from i to (i, id), where id is an instance identifier. Equality check on id can be done via IBE, hence this generalized direct sum can be embedded into $P^{\mathsf{IBE}} \wedge \mathsf{DS}[\mathcal{P}]$. This is also similar to an idea for implementing the Okamoto-Takashima variant of ABE definition, discussed in §9.2.

On Casting Known PESs. Any existing PESs proven secure in the sense of perfect or computational master-key hiding [7,12,8] are *not trivially broken* in the sense of [2]. Hence these PESs are also automatically symbolically secure via the first theorem in [2]. Therefore, we can cast them all here and use as basic PESs for basic predicates to be composed into our dynamic transformations. To name just a few, we can cast the known PES constructions [7,12,8] for predicates of doubly-spatial encryption, regular languages, branching programs, range/subset membership [10], revocation [43].

Static Compositions of Predicates. Static compositions of predicates, where the policy over predicate is fixed at setup, can be obtained from our dynamic transformations by simply fixing the policy. For any P_1, P_2, its conjunction, $P_1 \wedge P_2$, can be embedded into $\mathsf{KP}_{\mathsf{AND}}[P_1 \odot P_2]$ (or $\mathsf{CP}_{\mathsf{AND}}[P_1 \odot P_2]$). The same applies for the disjunction, $P_1 \vee P_2$.

Example of ABE with Multi-layer Functionality: Fine-grained Tax Return Audit System. We can consider ABE where a lower layer is a regular expression matching, while an upper one is a formula matching. Waters [41] suggests a tax return audit system as a motivating application of his ABE for Regular Language (RL). There, one considers a set of tax-return documents, each consists of a public deduction claim and a private data which is the amount of tax deductions. If the claim matches a certain regular expression (e.g., according to tax laws), the private data can be read by an auditor who has a key related to that expression. ABE for RL is needed as a claim can be arbitrarily long. This basic system can be considered as an access control system over claims (which can be considered as *objects*). In real-world situations, however, we may also want access control over auditors (which can be considered as *subjects*), *on top of the basic system*. As motivated by recent real-world leak cases like the Panama or Paradise Papers, such a fine-grained control of auditors (instead of all-or-nothing) is much needed. (We want to protect data against these insider auditors who might turn malicious and leak files.) This is possible via our $\mathsf{CP}[P^{\mathsf{RL}}]$. It was not possible before as we did not know how to combine RL under another layer of policy. Similar to the audit system, regular expressions can be used for defining rules in firewall systems or scanning virus for webpages, as suggested also in [41], hence our framework can produce "fine-grained" versions of such systems.

L.4 Possible Real-world Applications

Geographically Distributed Key Management. In the talk in the Real-world cryptography conference 2018, Sullivan [37] (Cloudfare inc.) proposed a geographically distributed key management scheme (geo manager) that is very useful for building efficient and secure communication over high-latency network, possibly across continents of the world. His scheme includes a building block which is essentially the AND composition of IBBE and IBR, in our terminology. His scheme is implemented by AND secret sharing the message as $M = M_1 \oplus M_2$ and encrypting each with IBBE of Delerablée [22] and IBR of Attrapadung *et al.* [11]. However, it is completely broken via a collusion attack (this was quickly pointed out by an audience after the talk).

Our framework provide a clean and modular solution for the above application by composing IBBE and IBR in our framework. Concrete pair encodings for IBBE, IBR can be promptly obtained from §C. In particular, the end product, namely, fully secure ABE, is inherently collusion-secure. Moreover, due to the dynamic nature of our composition framework, it is also flexible to do even more than the AND of IBBE/IBR, if needed so in the future. This can be done without any cost (only changing a policy over IBBE/IBR, or even mixing with other predicates).

L.5 Some New Insight from Our Framework

An informal discussion here is supplementary to the end of §2. What we found interesting and might give some new insight (although we do not formalize here) is that all the symbolic proofs in this paper essentially contain "mask" vectors that encode exactly the necessary condition when a predicative machine M does not not accept an input Y. On one hand, this somewhat follows along the same line as recent works of Ambrona *et al.* [3] and Agrawal and Chase [2] (which we base on), in the sense that their proofs give some "witnesses" (as used in [3]) or "certificates" (as used in [2]) that act as a short proof that such a scheme is secure. In symbolic security, these refer to those substituted vectors/matrices for a pair (M, Y) where M does not accept Y. However, finding appropriate substituted vectors/matrices are somewhat not trivial tasks (as stated in [2]). On the other hand, in our proofs, non-acceptance conditions for the considering predicate (which should definitely be exploited in order to deduce a security proof) are encoded in vectors in a somewhat more explicit manner. This is inherently due to the fact that our transformations are modular and abstract, hence we can separate what values come from based predicates or the predicative machine itself. This is in contrast with all the symbolic proofs in [2] which were designed specifically and monolithically to each PES, and hence all components (whether from based predicates or policy over them) are intertwined. Particularly, we think that the proof for our predicative DFA scheme decouples essential mechanisms in the selective security proof in Waters' DFA-based ABE [41] and in the selective and co-selective master-key hiding proofs in Attrapadung's DFA-based ABE [7], in an abstract manner.

L.6 More Related Works and Open Problems

This subsection provides more related works and some open problems, deferred from §1.

More Related Works. Conditional Disclosure of Secret (CDS) is a related primitive that can be viewed as a limited form of private-key attribute-based encryption which offers one-time information-theoretic security. Applebaum *et al.* [5] studied generic compositions of CDS schemes. The composition takes CDS schemes for predicate h_i, and a boolean function g as inputs, and outputs a CDS scheme for predicate $g(h_1, \ldots, h_n)$. Since the composition formula g is fixed for the derived scheme, we can view their composition as a static one. It might be interesting to define what dynamic composition means in the context of CDS and construct such compositions.

Ambrona *et al.* [4] recently built an automated tool for checking symbolic security of the class of ABE called rational-fraction induced ABE. It might be interesting to see how the tools can be used for deriving automated proofs for our schemes for composed predicates. In their case studies for the (bounded) ABE scheme of [24], the automated tool is able to prove security for only small fixed-size attribute sets and policies. Since our composed predicates are even more complex, it might require some breakthroughs to be able to derive a fully automated tool that works for our schemes.

Open Problem. It is an open problem to construct a fully secure ABE scheme for dynamically composed predicates, *e.g.,* $\mathsf{KP}[\mathcal{P}]$—the key-policy-span-program-augmented predicate over a set $\mathcal{P}$ of predicates, *under static (non-q-type) assumptions*, such as DLIN. It would require some breakthroughs to construct one. This is since even in the simplest setting where we consider the policy-augmented predicate over *one predicate* P and *one input*, namely, $\mathsf{KP1}[P]$, we already require the property of unbounded attribute multi-use in one policy and the property of unbounded-size policy; and no such ABE (which achieves both unbounded properties and is fully secure under DLIN) is known, even for the specific predicates $\mathsf{KP1}[P^{\mathsf{IBBE}}]$ (*i.e.,* completely unbounded KP-ABE for MSP). We note that there are only two available fully secure KP-ABE schemes for MSP that achieve the unbounded attribute multi-use property under DLIN, namely, the schemes by Takashima [38] and Kowalczyk *et al.* [26]. However, both schemes pose restrictions on bounded-size policies. Put in other words, solving the problem of constructing fully secure completely unbounded KP-ABE schemes for MSP under DLIN, *even without compositions*, would already require some breakthroughs; the problem of constructing ABE for composed predicates among them under DLIN should require even more new techniques to accomplish. All in all, it is thus natural to use a stronger assumption (the q-ratio assumption) for now.

Table of Contents

1 Introduction 1
2 Intuition and Informal Overview 6
3 Preliminaries 12
3.1 Definitions for General ABE 13
3.2 Pair Encoding Scheme Definition 13
3.3 Symbolic Property of PES 14
3.4 Definitions for Some Previous Predicates 15
4 Admissible Pair Encodings 16
5 Ciphertext-policy Augmentation 17
6 Key-policy Augmentation 21
7 Direct Sum and Augmentation over Predicate Set 22
7.1 Simple Direct Sum by Parameter Concatenation 23
7.2 Efficient Direct Sum with Parameter Reuse 24
8 Predicative Automata 25
9 Applications 28
9.1 ABE for New Predicates 28
9.2 Revisiting Known Predicates 28
References 30
A Supplementary Table and Figure 33
B Recapped Known Definitions and Results 33
B.1 Security Definition for ABE 33
B.2 Main Theorem in the Agrawal-Chase Framework 33
B.3 Embedding Lemma 35
C Basic Concrete Pair Encoding Schemes 35
C.1 Basic Schemes 35
C.2 Basic Admissible Schemes 36
D PES for KP-ABE for Non-monotone Span Programs 37
E Proof for Layer Transformation 40
F Verifying Proof for Ciphertext-policy Augmentation 40
G Definition and Proof for Key-policy Augmentation 43
G.1 Definition 43
G.2 Proof for Key-policy Augmentation 43
H Proofs for Direct Sum 47
H.1 Proof for Parameter Concatenation Scheme 48
H.2 Proof for Parameter Reuse Scheme 49
I Proof for Predicative Automata 50
I.1 Properties for Predicative DFA 51
I.2 Summary for Polynomials/Variables in Our Construction 52
I.3 Proof for Our Predicative DFA Scheme 53
J Proof for Predicative Branching Program 72
K Unbounded Arbitrary Mixed-policy Augmentation 74
L Further Discussions 76
L.1 On Difficulty of ABE for Non-monotone Span Programs 76
L.2 Remark on the q-ratio Assumption 76
L.3 More Applications and Extensions 77

L.4 Possible Real-world Applications 78
L.5 Some New Insight from Our Framework 78
L.6 More Related Works and Open Problems 79