Dual System Encryption via Doubly Selective Security: Framework, Fully-secure Functional Encryption for Regular Languages, and More

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

Abstract

Dual system encryption techniques introduced by Waters in Crypto'09 are powerful approaches for constructing fully secure functional encryption (FE) for many predicates. However, there are still some FE for certain predicates to which dual system encryption techniques seem inapplicable, and hence their fully-secure realization remains an important problem. A notable example is FE for regular languages, introduced by Waters in Crypto'12.

We propose a generic framework that abstracts the concept of dual system encryption techniques. We introduce a new primitive called *pair encoding* scheme for predicates and show that it implies fully secure functional encryption (for the same predicates) via a generic construction. Using the framework, we obtain the first fully secure schemes for functional encryption primitives of which only selectively secure schemes were known so far. Our three main instantiations include FE for regular languages, unbounded attribute-based encryption (ABE) for large universes, and ABE with constant-size ciphertexts.

Our main ingredient for overcoming the barrier of inapplicability for the dual system techniques to certain predicates is a computational security notion of the pair encoding scheme which we call *doubly selective security*. This is in contrast with most of the previous dual system based schemes, where information-theoretic security are implicitly utilized. The doubly selective security notion resembles that of selective security and its complementary notion, co-selective security, and hence its name. Our framework can be regarded as a method for boosting doubly selectively security (of encoding) to full security (of functional encryption).

Besides generality of our framework, we remark that improved security is also obtained, as our security proof enjoys tighter reduction than previous schemes, notably the reduction cost does not depend on the number of all queries, but only that of *pre-challenged* queries.

Keywords. Dual system encryption, Functional encryption for regular languages, Attribute-based encryption, Constant-size ciphertexts, Full security, Unified framework, Tighter reduction.

An extended abstract of this paper appears in the proceedings of Eurocrypt 2014. This is the full version.

1 Introduction

Dual system encryption techniques introduced by Waters [37] have been successful approaches for proving adaptive security (or called full security) for functional encryption (FE) schemes that are based on bilinear groups. These include adaptively-secure schemes for (hierarchical) id-based encryption (HIBE) [37, 23, 25, 22], attribute-based encryption (ABE) for Boolean formulae [27, 32, 26], inner-product encryption [27, 32, 1, 33, 34], and spatial encryption [1, 19].

Due to structural similarities between these fully secure schemes obtained via the dual system encryption paradigm and their selectively secure counterparts previously proposed for the same primitive[1], it is perhaps a folklore that the dual system encryption approach can somewhat elevate the latter to achieve the former. This is unfortunately not so, or perhaps not so clear, as there are some functional encryption schemes that are only proved selectively secure at the present time and seem to essentially encounter problems when applying dual system proof techniques. A notable example is FE for regular languages proposed by Waters [38], for which fully secure realization remains an open problem.

In this paper, we affirmatively solve this by proposing the first fully secure functional encryption for regular languages. Towards solving it, we provide a generic framework that captures the core concept of the dual system encryption techniques. This gives us an insight as to why it was not clear in the first place that dual system encryption techniques can be successfully applied to certain primitives, but not others. Such an insight leads us not only to identify the obstacle when applying the techniques and then to find a solution that overcomes it, but also to improve the performance of security proofs in a generic way. Namely, our framework allows tighter security reduction.

We summarize our contributions below. We first recall the notion of functional encryption, formulated in [7]. Well-known examples of functional encryption such as ABE and the more recent one for regular languages can be considered as "public-index" predicate encryption, which is a class of functional encryption. We focus on this class in this paper.[2] A primitive in this class is defined by a predicate R. In such a scheme, a sender can associate a ciphertext with a ciphertext attribute Y while a secret key is associated with a key attribute X. Such a ciphertext can then be decrypted by such a key if $R(X, Y)$ holds.

1.1 Summary of Our Contributions

Main Contributions. In this paper, we propose a generic framework that captures the concept of dual system encryption techniques. It is generic in the sense that it can be applied to *arbitrary* predicate R. The main component in our framework is a new notion called *pair encoding* scheme defined for predicate R. We formalize its security properties into two notions called *perfectly master-key hiding*, which is an information-theoretic notion, and *doubly selectively master-key hiding*, which is a computational notion. The latter consists of two notions which are *selective master-key hiding* and its complementary one called *co-selective master-key hiding* (and hence is named *doubly*). Our main results are summarized as follows.

- **Generic Construction.** We construct a generic construction of fully secure functional encryption for predicate R from any pair encoding scheme for R which is either perfectly master-key hiding or doubly selectively master-key hiding. Our construction is based on composite-order bilinear groups.

[1] One explicit example is the fully secure HIBE of Lewko and Waters [23], which has the structure almost identical to the selectively secure HIBE by Boneh, Boyen, Goh [5].

[2] In this paper, the term "functional encryption" refers to this class.

- **Instantiations.** We give concrete constructions of pair encoding schemes for notable three predicates of which there is no known fully-secure functional encryption realization. By using the generic construction, we obtain fully secure schemes. These include the following.

  - The first fully-secure functional encryption for regular languages. Only a selectively-secure scheme was known [38]. We indeed improve not only security but also efficiency: ours will work on *unbounded alphabet* universe, as opposed to *small universe* as in the original construction.
  - The first fully-secure unbounded key-policy ABE with large universes. Such a system requires that no bound should be posed on the sizes of attribute set and policies. The available schemes are either selectively-secure [25, 29] or small-universe [26] or restricted for multi-use of attributes [34].
  - The first fully-secure key-policy ABE with constant-size ciphertexts (and with large universes). The available schemes are either only selectively-secure scheme [2], or restricted to small classes of policies [9].

  Our three underlying pair encoding schemes are proved doubly selectively secure under new static assumptions, each of which is parameterized by the sizes of attributes in one ciphertext or one key, but *not* by the number of queries. These can be considered comparable to those assumptions for the respective selectively secure counterparts ([38, 29, 2], resp.).

- **Improved Security Reduction.** By starting from a pair encoding scheme which is doubly selectively master-key hiding, the resulting functional encryption can be proved fully secure with tighter security reduction to subgroup decision assumptions (and the doubly selective security). More precisely, it enjoys reduction cost of $O(q_1)$, where q_1 is the number of *pre-challenged* key queries. This improves all the previous works based on dual system encryption (except only one recent work on IBE by [8]) of which reduction encumbers $O(q_{\text{all}})$ security loss, where q_{all} is the number of *all* key queries. As an instantiation, we propose an IBE scheme with $O(q_1)$ reduction, while enjoys similar efficiency to [23].

More Contributions. We also obtain the following results.

- **Dual Scheme Conversion.** We obtain a generic and simple conversion that can convert any encoding scheme for a predicate to another for its *dual* predicate, where the role of of ciphertext and key attributed are swapped. (For example, key-policy ABE and ciphertext-policy ABE are dual to each other). We show that if the former is *perfectly secure*, then so is the latter, generically. In the case of *doubly selectively secure* encoding, we do not have generic implication. To this end, we directly construct an encoding scheme for the dual FE for regular languages and prove its doubly selective security. This implies the first fully secure dual FE for regular languages.

- **Unified Treatment for Existing Constructions.** We can cast many existing FE constructions as special cases of our framework by indicating their corresponding encoding schemes. These include Lewko-Waters IBE [23], Lewko et al. small-universe ABE (both key-policy and ciphertext-policy) [27], doubly spatial encryption [19], negated spatial encryption [1]. While the former two [23, 27] were already fully-secure, the latter two [19, 1] were proved only selectively-secure in their corresponding original papers. We show that the encodings extracted from these constructions are indeed perfectly-secure. Hence, via our framework we immediately obtain new fully-secure schemes of the latter two. In particular, we obtain the first fully-secure negated spatial encryption (to the best of our knowledge). We note that a fully-secure doubly-spatial encryption scheme was also given in [10].

- **New Improved Constructions.** We present some improvements to ABE of Lewko et al. [27].
  - New ABE (for small universe) with improved efficiency. By a simple observation, we tweak the perfectly secure encoding scheme extracted from [27] and obtain a new KP-ABE with key size reduced to half and a new CP-ABE with ciphertext size reduced to half, for free.
  - New ABE for large universe. This is also for free (no new assumption is needed), as we propose its corresponding encoding scheme that achieves perfect security. Fully-secure ABE schemes for large universe were already given in [32], but these have slightly different semantics from the original definition of ABE provided by [18]. (The difference was discussed in [2]).
- **New Primitives.** We propose a new primitive called key-policy over doubly spatial encryption. It generalizes the key-policy ABE in such a way that attributes are generalized to affine spaces, and the equality relation of two attributes is generalized to the doubly spatial relation [19], which amounts to check if two affine spaces intersect. This encompasses ABE and doubly spatial encryption into one notion. Indeed, our two main instantiations for ABE above (unbounded, or constant-size ciphertext) are obtained as special cases of this.

1.2 Perspective

Our framework can be considered as a "toolbox" for checking whether dual system techniques can be applied in a naive (or called classical) way or not, and if not, how one can overcome the barrier. More precisely, we propose the following procedure. One would start with a selectively-secure functional encryption scheme that is compatible with pair encoding syntax. We first extract the pair encoding scheme from it (this should be an easy procedure).

- **Checking Applicability.** If the encoding scheme is not *perfectly* master-key hiding, then dual system encryption techniques cannot be applied in a classical way. Intuitively, this is due to the fact that a core technique for dual system proofs requires some information-theoretic argument to hold, and it is exactly this perfect security of encoding that captures such an argument.
- **Overcoming the Barrier.** We overcome the obstacle from the information-theoretic argument by using *computational* security instead, and thus utilizing *doubly selective security* of pair encoding. Proving this security for encoding is usually harder than the perfect security, since it would require a reduction to computational assumptions. Fortunately, due to the definitional similarities for selective security of the pair encoding to selective security of the starting functional encryption scheme, we can roughly reuse the whole proof strategy! The harder part would be to prove co-selective security. But this, again, can be borrowed from selective security of its *dual predicate*. This computational technique was implicitly used by Lewko and Waters [26] specifically for their ABE construction. We generalize to work for any predicate.

We indeed use this procedure to construct our three main instantiations. We briefly explain for the case of fully-secure FE for regular languages here. We first extract an encoding scheme from Waters' selectively secure scheme [38], and prove that it is not perfectly master-key hiding. We then modify the encoding (one of the techniques for modifying is explained in Remark 4) and prove its security by borrowing strategy from Waters' proof. Now the hardest part is to prove co-selective security. We use completely new techniques since there was no known dual scheme of Waters'.

1.3 Related Work

Chen and Wee [8] recently proposed the notion of dual system groups. It can be seen as a *complementary* work to ours: their construction unifies group structures where dual system techniques are applicable (namely, composite-order and prime-order groups) but for specific primitives (namely, IBE and HIBE), while our construction unifies schemes for *arbitrary predicate* but over specific groups (namely, composite-order bilinear groups). It is also worth mentioning that the topic of functional encryption stems from many research papers: we list some more here [3, 6, 18, 20, 31, 36]. Recent results give very general FE primitives such as ABE or FE for circuits [17, 13, 15, 14], and for Turing Machines [16], but most of them might still be considered as proofs of concept, since underlying cryptographic tools such as multilinear maps [12] seem still inefficient. Constructing fully secure ABE for circuits *without complexity leveraging* is an open problem.

Concurrent and Independent Work. Concurrently and independently, Wee [39] recently proposed the notion called predicate encoding. This is similar to one of our notions, namely the notion of *perfectly-secure* pair encoding, which abstracts the classical dual system techniques. Furthermore, his instantiations of KP-ABE, CP-ABE (for small universe) with improved efficiency, and doubly spatial encryption are similar to ours (Scheme 9, 11, 14, respectively, in §9). Our negated spatial encryption (Scheme 15 in §9.2) is also related to his non-zero inner-product scheme (*cf.* [1]).

1.4 Organization of the Paper

The paper can be divided into two parts: framework and instantiations. Intuition for the framework part is fully explained in the overview in §2, while intuition for each instantiation is given in its corresponding section. We start the main body by giving some definitions and notations in §3. We then formally describe the main framework in §4. The framework for dual scheme conversion is given later in §8.1. The remaining sections are for instantiations. We begin with the simplest one, which is IBE with tighter reduction, in §5. The security proof for the encoding of this IBE is a base to the proofs for other instantiations in the paper; hence, it might be useful to read this before going directly to others. We present our FE for regular languages in §6, and its dual scheme later in §8.2. We present unbounded ABE in §7.1, ABE with constant-size ciphertexts in §7.2, and their generalized primitive in §7.3. We demonstrate how our framework unifies existing schemes and improves them in §9. Postponed proofs are in the appendix. The reference and the table of contents are provided at the end.

2 An Intuitive Overview of Our Framework

In this section, we provide an intuition for our formalization of the dual system techniques and describe how we define pair encoding schemes. In our framework, we view a ciphertext $(\boldsymbol{C}, C_0)$ (encrypting M), and a key $\boldsymbol{K}$ as

$$\boldsymbol{C} = g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}, \quad C_0 = Me(g_1, g_1)^{\alpha s}; \qquad\qquad \boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})}$$

where $\boldsymbol{c}$ and $\boldsymbol{k}$ are *encoding functions* of attributes Y, X associated to ciphertext and key, respectively. The bold font represents vectors. Our aim is to formalize such functions by providing sufficient conditions so that the scheme can be proved fully-secure in a generic way. We call such functions *pair encoding* for predicate R, since they encode a pair of attributes which are inputs to predicate R. They can be viewed as (multi-variate) polynomials in variables from $\boldsymbol{s}$ (which includes s), $\boldsymbol{h}, \boldsymbol{r}$, and α. Intuitively, α corresponds to a master key, $\boldsymbol{h}$ corresponds to parameter that will define public key $g_1^{\boldsymbol{h}}$, and $\boldsymbol{s}, \boldsymbol{r}$ correspond to randomness in ciphertexts and keys, respectively. We would require the following: (1) *correctness*, stating that if $R(X, Y) = 1$ then both encoding functions can be paired to obtain αs; and (2) *security*, which is the property when $R(X, Y) = 0$, and we show how to define it below. The key novelty of our abstraction stems from the way we define

Table 1: Summary for properties used in each transition for $\boldsymbol{C}, \boldsymbol{K}$.

Transition	Changes in $\mathbb{G}_{p_2}$	Indistinguishability under	Other properties of pair encoding
$\boldsymbol{C} : 0 \to 1$	$g_2^{\boldsymbol{c}(\mathbf{0},\mathbf{0})} \to g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})}$	subgroup decision	linearity, param-vanishing
$\boldsymbol{K} : 0 \to 1$	$g_2^{\boldsymbol{k}(0,\mathbf{0},\mathbf{0})} \to g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}$	subgroup decision	linearity, param-vanishing
$\boldsymbol{K} : 1 \to 2$	$g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \to g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}$	security of encoding	none
$\boldsymbol{K} : 2 \to 3$	$g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \to g_2^{\boldsymbol{k}(\hat{\alpha},\mathbf{0},\mathbf{0})}$	subgroup decision	linearity, param-vanishing

the security of encoding. Along the discussion, for a better understanding, a reader may think of the equality predicate and the Boneh-Boyen [4] IBE as a concrete example. Their encoding would be: $\boldsymbol{c}(s, \boldsymbol{h}) = (s, s(h_1 + h_2 Y))$ and $\boldsymbol{k}(\alpha, r, \boldsymbol{h}) = (\alpha + r(h_1 + h_2 X), r)$, where $\boldsymbol{h} = (h_1, h_2)$.

We first recall how dual system encryption techniques can be used to achieve *adaptive security*. The idea is to mimic the functionality of the encryption scheme in the *semi-functional* space, and to define the corresponding parameter $\hat{\boldsymbol{h}}$ in the semi-functional space to be independent from that of normal space, $\boldsymbol{h}$. Adaptive security is then obtained by observing that $\hat{\boldsymbol{h}}$ will not appear anywhere until the first query, which means that the reduction algorithm in the proof can adaptively deal with the adversary since it does not have to fix $\hat{\boldsymbol{h}}$ in advance. This is in contrast with $\boldsymbol{h}$, which is fixed in the public key $g_1^{\boldsymbol{h}}$. In the case of composite-order groups, the semi-functional space is implemented in a subgroup $\mathbb{G}_{p_2}$ of a group $\mathbb{G}$ of composite order $p_1 p_2 p_3$ (and the normal space is in $\mathbb{G}_{p_1}$).

Our purpose of abstraction is to capture the above mechanism in a generic way, while at the same time, to incorporate the security of encoding. Our main idea for doing this is to define semi-functional types of ciphertexts and keys explicitly *in terms of pair encoding functions*, so that the scheme structure would be copied to the semi-functional space. More precisely, we define semi-functional ciphertexts and keys as follows: C_0 is unmodified, and let

$$\boldsymbol{C} = \begin{cases} g_1^{\boldsymbol{c}(s,\boldsymbol{h})} \cdot g_2^{\boldsymbol{c}(\mathbf{0},\mathbf{0})} & \text{(normal)} \\ g_1^{\boldsymbol{c}(s,\boldsymbol{h})} \cdot g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})} & \text{(semi)} \end{cases}, \qquad \boldsymbol{K} = \begin{cases} g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\mathbf{0},\mathbf{0})} & \text{(normal)} \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} & \text{(semi type 1)} \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} & \text{(semi type 2)} \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} & \text{(semi type 3)} \end{cases}$$

where '·' denotes the component-wise group operation. The "semi-functional variables" (those with the hat notation) are defined to be independent from the normal part. (We neglect mask elements from $\mathbb{G}_{p_3}$ now for simplicity).

We then recall that the proof strategy for the dual system techniques uses hybrid games that modifies ciphertexts and keys from normal to semi-functional ones, and proves indistinguishability between each transition. By defining semi-functional types as above, we can identify which transition uses *security of encoding* and which one uses *security provided by composite-order groups* (namely, subgroup decision assumptions). We provide these in Table 1. In particular, we identify that the security of encoding is used in the transition from type 1 to type 2 semi-functional keys. We note that how to identify this transition was unclear in the first place, since in all the previous dual system based schemes (to the best of our knowledge), the indistinguishability of this form is *implicitly* employed inside another transition (*cf.* nominally semi-functional keys in [27]).

We explore both types of transitions and define properties needed, as follows.

Table 2: Summary of approaches for defining the security of encoding.

Indistinguishability between		Security	Implicit in
$\left\{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}}),\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})\right\}$	$,\left\{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}}),\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})\right\}$	info-theoretic	all but [26, 8]
$\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})},g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}\right\}$	$,\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})},g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}\right\}$	computational	[26]
$\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})},\{g_2^{\boldsymbol{k}_i(0,\hat{\boldsymbol{r}}_i,\hat{\boldsymbol{h}})}\}_{i\in Q}\right\}$	$,\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})},\{g_2^{\boldsymbol{k}_i(\hat{\alpha},\hat{\boldsymbol{r}}_i,\hat{\boldsymbol{h}})}\}_{i\in Q}\right\}$	computational	new

Transition Based on the Security of Encoding. We simply define the security of encoding to be just as what we need for the transition definition. More precisely, the security of encoding (in the "basic" form) requires that, if $R(X,Y)=0$, then the following distributions are indistinguishable:

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

where the probability taken over random $\hat{\boldsymbol{h}}$ (and others). We remark a crucial point that the fact that we define keys of *normal* types and semi-functional *type 3* to not depend on $\hat{\boldsymbol{h}}$ allows us to focus on the distribution corresponding to only *one key at a time*, while "isolating" other keys. (This is called key isolation feature in [26]). We provide more flavors of the definition below. Indeed, the computational variant is what makes our framework powerful.

Transitions Based on Subgroup Decision Assumptions. We require *all* pair encoding schemes to satisfy some properties in order to use subgroup decision assumptions. We identify the following two properties: *parameter-vanishing* and *linearity*.

$$\textbf{(Param-Vanishing)} \qquad \boldsymbol{k}(\alpha,\boldsymbol{0},\boldsymbol{h})=\boldsymbol{k}(\alpha,\boldsymbol{0},\boldsymbol{0}).$$

$$\textbf{(Linearity)} \qquad \boldsymbol{k}(\alpha_1,\boldsymbol{r}_1,\boldsymbol{h})+\boldsymbol{k}(\alpha_2,\boldsymbol{r}_2,\boldsymbol{h})=\boldsymbol{k}(\alpha_1+\alpha_2,\boldsymbol{r}_1+\boldsymbol{r}_2,\boldsymbol{h}),$$
$$\boldsymbol{c}(\boldsymbol{s}_1,\boldsymbol{h})+\boldsymbol{c}(\boldsymbol{s}_2,\boldsymbol{h})=\boldsymbol{c}(\boldsymbol{s}_1+\boldsymbol{s}_2,\boldsymbol{h}).$$

Linearity makes it possible to indistinguishably change the randomness between $\boldsymbol{0}$ and $\hat{\boldsymbol{r}}$ (in the case of $\boldsymbol{k}$), and between $\boldsymbol{0}$ and $\hat{\boldsymbol{s}}$ (in the case of $\boldsymbol{c}$) under subgroup decision assumptions, but without changing the other variables (*i.e.,* $\hat{\alpha},\hat{\boldsymbol{h}}$). Parameter-vanishing can then "delete" $\hat{\boldsymbol{h}}$ when $\hat{\boldsymbol{r}}=\boldsymbol{0}$. The latter makes it possible to obtain the key isolation, required for the previous type of transition. A subgroup decision assumption states that it is hard to distinguish if $t_2=0$ or $t_2 \xleftarrow{\$} \mathbb{Z}_{p_2}$ in $T=g_1^{t_1}g_2^{t_2}$. The intuition of how to use this assumption in conjunction with linearity is, for example, to simulate a key as $g_1^{\boldsymbol{k}(\alpha,\boldsymbol{0},\boldsymbol{h}')}T^{\boldsymbol{k}(0,\boldsymbol{r}',\boldsymbol{h}')}$, for known $\alpha,\boldsymbol{r}',\boldsymbol{h}'$ chosen randomly. This is a normal key if $t_2=0$ and semi-functional type-1 if $t_2 \xleftarrow{\$} \mathbb{Z}_{p_2}$. In doing so, we implicitly set $\boldsymbol{h}=\boldsymbol{h}'$ mod p_1 and $\hat{\boldsymbol{h}}$ $=\boldsymbol{h}'$ mod p_2, but these are independent exactly due to the Chinese Remainder Theorem. (The last property is referred as parameter-hiding in prior work). We also note that linearity implies homogeneity: $\boldsymbol{c}(\boldsymbol{0},\boldsymbol{0})=0,\boldsymbol{k}(0,\boldsymbol{0},\boldsymbol{0})=0$, and hence we can write the normal ciphertext and key as above.

Perfect Security of Pair Encoding. We identify three flavors for the security of encoding that imply the basic form of security defined above. We list them in Table 2. We refer the first notion as the *perfectly master-key hiding* security, which is an *information-theoretic* notion. All the previous dual system based schemes (except [26, 8]) implicitly employed this approach. For some esoteric predicates (*e.g.,* the regular language functionality), the amount of information from $\hat{\boldsymbol{h}}$ needed for hiding $\hat{\alpha}$ is not sufficient. This is exactly the reason why the "classical" dual system approach is inapplicable to FE for regular languages.

Computational Security of Pair Encoding. The second flavor (the second line of Table 2, which is exactly the same as the aforementioned basic form) employs *computational* security argument to hide $\hat{\alpha}$, and can overcome the obstacle of insufficient entropy, suffered in the first approach. This approach was introduced by Lewko and Waters [26] to overcome the obstacle of multi-use restriction in KP-ABE. We generalize their approach to work for any predicate.

When considering computational approaches, the ordering of queries from the adversary becomes important since the challenger is required to fix the value of $\hat{\boldsymbol{h}}$ after receiving the first query. This is reminiscent of the notion of *selective security* for FE, where the challenger would fix public parameters after seeing the challenge ciphertext attribute. To this end, we refer this notion as *selective* master-key hiding, if a query for Y (corresponding to the encoding $\boldsymbol{c}$) comes before that of X (for the encoding $\boldsymbol{k}$), and analogously, *co-selective* master-key hiding if a query for X comes before that of Y, where we recall that co-selective security [1] is a complementary notion of selective security.[3]

Tighter Reduction. The classical dual system paradigm requires $O(q_{\text{all}})$ transition steps, hence results in $O(q_{\text{all}})$ loss for security reduction, where q_{all} is the number of all key queries. This is since each step is reduced to its underlying security: subgroup indistinguishability or the security of encoding. This is the case for all the previous works except the IBE scheme of [8].[4] To overcome this obstacle, we propose the third flavor for security of encoding, shown in the third line of Table 2. This new approach is unique to our framework (no implicit use in the literature before). The idea is to observe that, for the selective security proof, the reduction can program the parameter once by using the information of the ciphertext attribute Y, and after that, *any* keys for X such that $R(X, Y) = 0$ can be produced. Therefore, we can organize all the *post-challenged* keys into the correlated distribution (hence, in Table 2, we set Q to be this set of queries). This has a great benefit since we can define a new type of transition where all these *post-challenged* keys are simultaneously modified from semi-functional type-1 to type-2 *all at once*, which results in tighter reduction, $O(q_1)$, where q_1 is the number of *pre-challenged* queries. On the other hand, one could try to do the same by grouping also all the *pre-challenged* queries and mimicking co-selective security, so as to obtain tight reduction (with $O(1)$ cost). However, this will not work since the parameter must be fixed already after *only the first query*.

3 Preliminaries, Definitions, and Notations

3.1 Functional Encryption

Predicate Family. We consider a predicate family $R = \{R_\kappa\}_{\kappa \in \mathbb{N}^c}$, for some constant $c \in \mathbb{N}$, where a relation $R_\kappa : \mathbb{X}_\kappa \times \mathbb{Y}_\kappa \to \{0, 1\}$ is a predicate function that maps a pair of key attribute in a space $\mathbb{X}_\kappa$ and ciphertext attribute in a space $\mathbb{Y}_\kappa$ to $\{0, 1\}$. The family index $\kappa = (n_1, n_2, \ldots)$ specifies the description of a predicate from the family.

Predicate in Different Domains. We mandate the first entry n_1 in κ to specify some domain; for example, the domain $\mathbb{Z}_N$ of IBE (the equality predicate), where we let $n_1 = N$. In what

[3] As a result, this also clarifies why [26] uses selective security techniques of KP-ABE and *CP-ABE* to prove the full security of KP-ABE. This is since selective security of an FE (CP-ABE, in their case) resembles co-selective security of its *dual* (KP-ABE).

[4] The IBE of [8] used a technique from Naor and Reingold [28] PRFs for their computational argument, which is different from ours.

follows, we will implement our scheme in composite-order groups and some relations among different domains in the same family will be used. We formalize them here. We omit κ and write simply R_N. We say that R is *domain-transferable* if for p that divides N, we have projection maps $f_1 : \mathbb{X}_N \to \mathbb{X}_p, f_2 : \mathbb{Y}_N \to \mathbb{Y}_p$ such that for all $X \in \mathbb{X}_N, Y \in \mathbb{Y}_N$:

- **Completeness.** If $R_N(X, Y) = 1$ then $R_p(f_1(X), f_2(Y)) = 1$.
- **Soundness.** (1) If $R_N(X, Y) = 0$ then $R_p(f_1(X), f_2(Y)) = 0$, *or* (2) there exists an algorithm that takes (X, Y) where (1) does not hold, and outputs a non-trivial factor F, where $p|F, F|N$.

The completeness will be used for correctness of the scheme, while the soundness will used in the security proof. All the predicates in this paper are domain-transferable. As an example, in the equality predicate (for IBE), R_N and R_p are defined on $\mathbb{Z}_N$ and $\mathbb{Z}_p$ respectively. The projective maps are simply modulo p. Completeness holds straightforwardly. Soundness holds since for $X \neq Y \pmod{N}$ but $X = Y \pmod{p}$, we set $F = X - Y$. The other predicates in this paper can be proved similarly and we omit them here.

Functional Encryption Syntax. A functional encryption (FE) scheme for predicate family R consists of the following algorithms.

- $\mathsf{Setup}(1^\lambda, \kappa) \to (\mathsf{PK}, \mathsf{MSK})$: takes as input a security parameter 1^λ and a family index κ of predicate family R, and outputs a master public key PK and a master secret key MSK.
- $\mathsf{Encrypt}(Y, M, \mathsf{PK}) \to \mathsf{CT}$: takes as input a ciphertext attribute $Y \in \mathbb{Y}_\kappa$, a message $M \in \mathcal{M}$, and public key PK. It outputs a ciphertext CT.
- $\mathsf{KeyGen}(X, \mathsf{MSK}, \mathsf{PK}) \to \mathsf{SK}$: takes as input a key attribute $X \in \mathbb{X}_\kappa$ and the master key MSK. It outputs a secret key SK.
- $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK}) \to M$: 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 indexes κ, all $M \in \mathcal{M}$, $X \in \mathbb{X}_\kappa$, $Y \in \mathbb{Y}_\kappa$ such that $R_\kappa(X, Y) = 1$. 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 Notion. A functional encryption scheme for predicate family R 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 $\mathcal{C}$. For our purpose of modifying games in next sections, we write some in the boxes. Let q_1, q_2 be the numbers of queries in Phase 1,2, respectively.

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

3.2 Bilinear Groups of Composite Order

In our framework, we consider bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3$, where p_1, p_2, p_3 are distinct primes, with an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. For our purpose, we define a bilinear group generator $\mathcal{G}(\lambda)$ that takes as input a security parameter λ and outputs $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3)$. For each $d|N$, $\mathbb{G}$ has a subgroup of order d denoted by $\mathbb{G}_d$. We let g_i denote a generator of $\mathbb{G}_{p_i}$. Any $h \in \mathbb{G}$ can be expressed as $g_1^{a_1}g_2^{a_2}g_3^{a_3}$, where a_i is uniquely determined modulo p_i. We call $g_i^{a_i}$ the $\mathbb{G}_{p_i}$ component of h. We recall that e has the bilinear property: $e(g^a, g^b) = e(g, g)^{ab}$ for any $g \in \mathbb{G}$, $a, b \in \mathbb{Z}$ and the non-degeneration property: $e(g, h) \neq 1 \in \mathbb{G}_T$ whenever $g, h \neq 1 \in \mathbb{G}$. In a bilinear group of composite order, we also have orthogonality: for $g \in \mathbb{G}_{p_i}, h \in \mathbb{G}_{p_j}$ where $p_i \neq p_j$ we have that $e(g, h) = 1 \in \mathbb{G}_T$. The Subgroup Decision Assumptions 1,2,3 [37, 23] and the 3DH assumption in a subgroup [26] are given below.

Definition 1 (Subgroup Decision Assumptions (SD))**.** Subgroup Decision Problem 1,2,3 are defined as follows. Each starts with $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$.

(SD1). Given $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, Z_3 \xleftarrow{\$} \mathbb{G}_{p_3}$, and $T \in \mathbb{G}$, decide if $T = T_1 \xleftarrow{\$} \mathbb{G}_{p_1p_2}$ or $T = T_2 \xleftarrow{\$} \mathbb{G}_{p_1}$.

(SD2). Let $g_1, Z_1 \xleftarrow{\$} \mathbb{G}_{p_1}, Z_2, W_2 \xleftarrow{\$} \mathbb{G}_{p_2}, Z_3, W_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Given g_1, Z_1Z_2, Z_3, W_2W_3, and $T \in \mathbb{G}$, decide if $T = T_1 \xleftarrow{\$} \mathbb{G}_{p_1p_2p_3}$ or $T = T_2 \xleftarrow{\$} \mathbb{G}_{p_1p_3}$.

(SD3). Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}$, $g_2, W_2, Y_2 \xleftarrow{\$} \mathbb{G}_{p_2}$, $Z_3 \xleftarrow{\$} \mathbb{G}_{p_3}$ and $\alpha, s \xleftarrow{\$} \mathbb{Z}_N$. Given $g_1, g_2, Z_3, g_1^{\alpha}Y_2, g_1^sW_2$, and $T \in \mathbb{G}_T$, decide if $T = T_1 = e(g_1, g_1)^{\alpha s}$ or $T = T_2 \xleftarrow{\$} \mathbb{G}_T$.

We define the advantage of an adversary $\mathcal{A}$ against Problem i for $\mathcal{G}$ as the distance $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD}i}(\lambda) := |\Pr[\mathcal{A}(\boldsymbol{D}, T_1) = 1] - \Pr[\mathcal{A}(\boldsymbol{D}, T_2) = 1]|$, where $\boldsymbol{D}$ denotes the given elements in each assumption excluding T. We say that the Assumption i holds for $\mathcal{G}$ if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD}i}(\lambda)$ is negligible in λ for any poly-time algorithm $\mathcal{A}$.

Definition 2 (3-Party Diffie Hellman Assumption, 3DH)**.** The 3DH Assumption in a subgroup assumes the hardness of the following problem: let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda), g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}, a, b, z \xleftarrow{\$} \mathbb{Z}_N$, given $\boldsymbol{D} = (g_2, g_2^a, g_2^b, g_2^z, g_1, g_3)$ and T, decide whether $T = g_2^{abz}$ or $T \xleftarrow{\$} \mathbb{G}_{p_2}$.

3.3 Notation

In general, we treat a vector as a row vector (written horizontally). Let $\mathbb{G}$ be a group. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{G}^n$. We denote $\boldsymbol{a} \cdot \boldsymbol{b} = (a_1 \cdot b_1, \ldots, a_n \cdot b_n)$, where '$\cdot$' is the group operation of $\mathbb{G}$. Note that it is the pairwise operation and not a dot product. For $g \in \mathbb{G}$ and $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{Z}^n$, we denote $g^{\boldsymbol{c}} = (g^{c_1}, \ldots, g^{c_n})$. Let $N \in \mathbb{N}$. Consider $M \in \mathbb{Z}_N^{d\times n}$ (the set of all $d \times n$ matrices in $\mathbb{Z}_N$). We denote its row space as $\mathsf{RowSp}(M) = \{\boldsymbol{w}M \mid \boldsymbol{w} \in \mathbb{Z}_N^d\}$. We denote the transpose of M as $M^\top$. We denote by g^M the matrix in $\mathbb{G}^{d\times n}$ of which its (i, j) entry is $g^{M_{i,j}}$, where $M_{i,j}$ is the (i, j) entry of M.

For $Q \in \mathbb{Z}_N^{\ell\times d}$, we denote $(g^Q)^M = g^{QM}$. Note that from M and $g^Q \in \mathbb{G}^{\ell\times d}$, we can compute g^{QM} without knowing Q, since its (i, j) entry is $\prod_{k=1}^{d}(g^{Q_{i,k}})^{M_{k,j}}$. (This will be used in §4.3). For $g^{\boldsymbol{c}}, g^{\boldsymbol{v}} \in \mathbb{G}^n$, we denote $e(g^{\boldsymbol{c}}, g^{\boldsymbol{v}}) = e(g, g)^{\boldsymbol{c}\boldsymbol{v}^\top} \in \mathbb{G}_T$.

4 Our Generic Framework for Dual-System Encryption

4.1 Pair Encoding Scheme: Syntax

In this section we formalize our main component: pair encoding scheme. It follows the intuition from the overview in §2. We could abstractly define it purely by the described properties; however, we opted to make a more concrete definition, which seems not to lose much generality (we discuss this below).

Syntax. A pair encoding scheme for predicate family R consists of four deterministic algorithms given by $\mathsf{P} = (\mathsf{Param}, \mathsf{Enc1}, \mathsf{Enc2}, \mathsf{Pair})$:

- $\mathsf{Param}(\kappa) \to n$. It takes as input an index κ and outputs an integer n, which specifies the number of *common variables* in $\mathsf{Enc1}$, $\mathsf{Enc2}$. For the default notation, let $\boldsymbol{h} = (h_1, \ldots, h_n)$ denote the the list of common variables.

- $\mathsf{Enc1}(X, N) \to (\boldsymbol{k} = (k_1, \ldots, k_{m_1});\ m_2)$. It takes as inputs $X \in \mathbb{X}_\kappa$, $N \in \mathbb{N}$, and outputs a sequence of polynomials $\{k_\iota\}_{\iota\in[1,m_1]}$ with coefficients in $\mathbb{Z}_N$, and $m_2 \in \mathbb{N}$ that specifies the number of its own variables. We require that each polynomial k_ι is a *linear combination of monomials* $\alpha, r_j, h_i r_j$, where $\alpha, r_1, \ldots, r_{m_2}, h_1, \ldots, h_n$ are variables.

  More precisely, it outputs a set of coefficients $\{b_\iota, b_{\iota,j}, b_{\iota,j,i}\}_{\iota\in[1,m_1], j\in[1,m_2], i\in[1,n]}$, which defines the sequence of polynomials $\{k_\iota \in \mathbb{Z}_N[\alpha, r_1, \ldots, r_{m_2}, h_1, \ldots, h_n]\}_{\iota\in[1,m_1]}$ where

$$k_\iota\Big(\alpha, (r_1, \ldots, r_{m_2}), (h_1, \ldots, h_n)\Big) = b_\iota \alpha + \left(\sum_{j\in[1,m_2]} b_{\iota,j} r_j\right) + \left(\sum_{\substack{j\in[1,m_2]\\ i\in[1,n]}} b_{\iota,j,i} h_i r_j\right).$$

- $\mathsf{Enc2}(Y, N) \to (\boldsymbol{c} = (c_1, \ldots, c_{w_1});\ w_2)$. It takes as inputs $Y \in \mathbb{Y}_\kappa$, $N \in \mathbb{N}$, and outputs a sequence of polynomials $\{c_\iota\}_{\iota\in[1,w_1]}$ with coefficients in $\mathbb{Z}_N$, and $w_2 \in \mathbb{N}$ that specifies the number of its own variables. We require that each polynomial c_ι is a *linear combination of monomials* $s, s_j, h_i s, h_i s_j$, where $s, s_1, \ldots, s_{w_2}, h_1, \ldots, h_n$ are variables.

  More precisely, it outputs a set of coefficients $\{a_\iota, a_{\iota,j}, a'_{\iota,i}, a_{\iota,j,i}\}_{\iota\in[1,w_1], j\in[1,w_2], i\in[1,n]}$, which defines the sequence of polynomials $\{c_\iota \in \mathbb{Z}_N[s, s_1, \ldots, s_{w_2}, h_1, \ldots, h_n]\}_{\iota\in[1,w_1]}$ where

$$c_\iota\Big((s, s_1, \ldots, s_{w_2}), (h_1, \ldots, h_n)\Big) = a_\iota s + \left(\sum_{j\in[1,w_2]} a_{\iota,j} s_j\right) + \left(\sum_{i\in[1,n]} a'_{\iota,i} h_i s\right) + \left(\sum_{\substack{j\in[1,w_2]\\ i\in[1,n]}} a_{\iota,j,i} h_i s_j\right).$$

- $\mathsf{Pair}(X, Y, N) \to \boldsymbol{E}$. It takes as inputs X, Y, N, and output $\boldsymbol{E} \in \mathbb{Z}_N^{m_1\times w_1}$.

Correctness. The correctness requirement is defined as follows.

- First, for any $N \in \mathbb{N}$, let $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, N)$, $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, N)$, and $\boldsymbol{E} \leftarrow \mathsf{Pair}(X, Y, N)$, we have that if $R_N(X, Y) = 1$, then

$$\boldsymbol{k}\boldsymbol{E}\boldsymbol{c}^\top = \alpha s$$

  where the equality holds symbolically. Note that since $\boldsymbol{k}\boldsymbol{E}\boldsymbol{c}^\top = \sum_{i\in[1,m_1], j\in[1,w_1]} E_{i,j} k_i c_j$, this correctness amounts to check if there is a linear combination of $k_i c_j$ terms summed up to αs.

- Second, for p that divides N, if we let $\mathsf{Enc1}(X,N) \to (\boldsymbol{k}; m_2)$ and $\mathsf{Enc1}(X,p) \to (\boldsymbol{k}'; m_2)$, then it holds that $\boldsymbol{k} \bmod p = \boldsymbol{k}'$. The requirement for $\mathsf{Enc2}$ is similar.

Remark 1. We mandate that the variables used in $\mathsf{Enc1}$ and those in $\mathsf{Enc2}$ are different except only those common variables in $\boldsymbol{h}$. We remark that in the syntax, all variables are only *symbolic*: no probability distributions have been assigned to them yet. (We will eventually assign these in the security notion and the generic construction). Note that m_1, m_2 can depend on X and w_1, w_2 can depend on Y. We also remark that each polynomial in $\boldsymbol{k}, \boldsymbol{c}$ has no constant terms.

Terminology. In what follows, we often omit N as input if the context is clear. We denote $\boldsymbol{k} = \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h})$ or $\boldsymbol{k}_X(\alpha, \boldsymbol{r}, \boldsymbol{h})$, and $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h})$ or $\boldsymbol{c}_Y(\boldsymbol{s}, \boldsymbol{h})$, where we let $\boldsymbol{h} = (h_1, \ldots, h_n), \boldsymbol{r} = (r_1, \ldots, r_{m_2}), \boldsymbol{s} = (s, s_1, \ldots, s_{w_2})$. We remark that s in $\boldsymbol{s}$ is treat as a special symbol among the others in $\boldsymbol{s}$, since it defines the correctness. We always write s as the first entry of $\boldsymbol{s}$. In describing concrete schemes in this paper in following sections, we often use symbols that deviate from the default notation (h_i, r_i, s_i in $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{s}$, respectively). In such a case, we will write $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{s}$ explicitly and omit writing the output m_2, w_2 since they merely indicate the sizes $m_2 = |\boldsymbol{r}|$, $w_2 = |\boldsymbol{s}| - 1$.

Remark 2. It is straightforward to prove that the syntax of pair encoding implies *linearity* and *parameter-vanishing*, symbolically. We opted to define the syntax this way (concrete, instead of abstract based on properties only) since for the generic construction (*cf.* §4.3) to work, we need one more property stating that $\boldsymbol{c}$ can be computed from $\boldsymbol{h}$ by a linear (or affine) transformation. This is for ensuring computability of ciphertext from the public key, since the public key will be of the form $g_1^{\boldsymbol{h}}$ and we can only do linear transformations in the exponent. This, together with linearity in $\boldsymbol{s}$, prompts to define linear-form monomials in $\mathsf{Enc2}$ as above. Contrastingly, there is no similar requirement for $\mathsf{Enc1}$; however, we define linear-form monomials similarly so that the roles of both encoding functions can be exchangeable in the dual scheme conversion in §8, where we will exchange the roles of the two encodings.

4.2 Pair Encoding Scheme: Security Definitions

Security. We define the security notions of pair encoding schemes as follows.

(Perfect Security). The pair encoding scheme P is *perfectly master-key hiding* (PMH) if the following holds. For $N \in \mathbb{N}$, if $R_N(X,Y) = 0$, let $n \leftarrow \mathsf{Param}(\kappa)$, $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X,N)$, $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y,N)$, then the following two distributions are identical:

$$\{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h}),\ \boldsymbol{k}(0,\boldsymbol{r},\boldsymbol{h})\} \qquad \text{and} \qquad \{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h}),\ \boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})\},$$

where the probability is taken over $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n, \alpha \xleftarrow{\$} \mathbb{Z}_N, \boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_N^{(w_2+1)}$.

(Computational Security). We define two flavors for computational security notions: *selectively secure* and *co-selectively secure master-key hiding* ($\mathsf{SMH}, \mathsf{CMH}$) in a bilinear group generator $\mathcal{G}$. We first define the following game template, $\mathsf{Exp}_{\mathcal{G},\mathsf{G},b,\mathcal{A}}(\lambda)$, for the flavor $\mathsf{G} \in \{\mathsf{CMH}, \mathsf{SMH}\}$, $b \in \{0,1\}$. It takes as input the security parameter λ and does the experiment with the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, and outputs b'. The game is defined as:

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

where st denotes the state information and the oracles $\mathcal{O}_1, \mathcal{O}_2$ in each state are defined below.

- **Selective Security.** $\mathcal{O}^1$ can be queried once while $\mathcal{O}^2$ can be queried polynomially many times.
  - $\mathcal{O}^1_{\mathsf{SMH},b,\alpha,\boldsymbol{h}}(Y^\star)$: Run $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star, p_2)$; $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_{p_2}^{(w_2+1)}$; return $\boldsymbol{C} \leftarrow g_2^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}$.
  - $\mathcal{O}^2_{\mathsf{SMH},b,\alpha,\boldsymbol{h}}(X)$: If $R_{p_2}(X, Y^\star) = 1$, then return $\perp$.

    Else, run $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, p_2)$; $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_{p_2}^{m_2}$; return $\boldsymbol{K} \leftarrow \begin{cases} g_2^{\boldsymbol{k}(0,\boldsymbol{r},\boldsymbol{h})} & \text{if } b = 0 \\ g_2^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} & \text{if } b = 1 \end{cases}$

- **Co-selective Security.** Both $\mathcal{O}^1, \mathcal{O}^2$ can be queried once.
  - $\mathcal{O}^1_{\mathsf{CMH},b,\alpha,\boldsymbol{h}}(X^\star)$: Run $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X^\star, p_2)$; $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_{p_2}^{m_2}$; return $\boldsymbol{K} \leftarrow \begin{cases} g_2^{\boldsymbol{k}(0,\boldsymbol{r},\boldsymbol{h})} & \text{if } b = 0 \\ g_2^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} & \text{if } b = 1 \end{cases}$
  - $\mathcal{O}^2_{\mathsf{CMH},b,\alpha,\boldsymbol{h}}(Y)$: If $R_{p_2}(X^\star, Y) = 1$, then return $\perp$.

    Else, run $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, p_2)$; $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_{p_2}^{(w_2+1)}$; return $\boldsymbol{C} \leftarrow g_2^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}$.

We call queries as *one-key*, *one-ciphertext* to emphasize when they can be asked once in the games.

We define the advantage of $\mathcal{A}$ in the security game $\mathsf{G} \in \{\mathsf{SMH}, \mathsf{CMH}\}$ for bilinear group generator $\mathcal{G}$ as $\mathsf{Adv}^{\mathsf{G}}_{\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}_{\mathcal{G},\mathsf{G},0,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathcal{G},\mathsf{G},1,\mathcal{A}}(\lambda) = 1]|$. We say that the pair encoding scheme P is selectively (resp., co-selectively) master-key hiding in $\mathcal{G}$ if $\mathsf{Adv}^{\mathsf{SMH}}_{\mathcal{A}}(\lambda)$ (resp., $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A}}(\lambda)$) is negligible for all all polynomial time attackers $\mathcal{A}$. If both hold, we say that it is *doubly selectively master-key hiding.*

Remark 3. We observe that, in contrast with usual security notions of public-key primitives, where some public keys would be given to the adversary, the notion of encoding is essentially *private-key* one. Hence, in particular, $g_2^{\boldsymbol{h}}$, which defines the parameter $\boldsymbol{h}$, needed *not* be sent to the adversary. Intuitively, this is since the master-key hiding security will be employed in the semi-functional space ($\mathbb{G}_{p_2}$), and the parameter then corresponds to *semi-functional parameter* $\hat{\boldsymbol{h}}$, which needs not be sent. (See explanation in §2). As a consequence, we can observe later that in all the proofs of computational master-key hiding security for instantiations, a simulator needs *not* explicitly compute $g_2^{\boldsymbol{h}}$, but can program it *implicitly* so as to just have consistency among all queries.

4.3 Generic Construction for Predicate Encryption from Pair Encoding

Construction. From a pair encoding scheme P for predicate R, we construct a functional encryption scheme for R, denoted $\mathsf{FE}(\mathsf{P})$, as follows.

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

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

$$\boldsymbol{C} = g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})} \in \mathbb{G}^{w_1}, \qquad C_0 = (e(g_1, g_1)^\alpha)^s M \in \mathbb{G}_T. \tag{1}$$

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

- $\mathsf{KeyGen}(X, \mathsf{MSK}, \mathsf{PK})$: Upon input $X \in \mathbb{X}_N$, run $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, N)$. Parse $\mathsf{MSK} = \alpha$. Recall that $m_1 = |\boldsymbol{k}|$. Pick $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$. Output the secret key SK:
$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot \boldsymbol{R}_3 \in \mathbb{G}^{m_1}. \tag{2}$$

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

Correctness. For $R_N(X, Y) = 1$, we have $R_{p_1}(X, Y) = 1$ from domain-transferability. Then,
$$e(\boldsymbol{K}^{\boldsymbol{E}}, \boldsymbol{C}) = e((g_1^{\boldsymbol{k}} \cdot \boldsymbol{R}_3)^{\boldsymbol{E}}, g_1^{\boldsymbol{c}}) = e(g_1, g_1)^{\boldsymbol{k}\boldsymbol{E}\boldsymbol{c}^\top} = e(g_1, g_1)^{\alpha s},$$
where the last equality comes from the correctness of the pair encoding scheme.

Semi-Functional Algorithms. These will be used in the proof only.

- $\mathsf{SFSetup}(1^\lambda, \kappa)$: This is exactly the same as $\mathsf{Setup}(1^\lambda, \kappa)$ except that it additionally outputs a generator $g_2 \xleftarrow{\$} \mathbb{G}_{p_2}$ and $\hat{\boldsymbol{h}} \xleftarrow{\$} \mathbb{Z}_N^n$. We call $\hat{\boldsymbol{h}}$ a semi-functional parameter.

- $\mathsf{SFEncrypt}(Y, M, \mathsf{PK}, g_2, \hat{\boldsymbol{h}})$: Upon inputs Y, M, PK, g_2 and $\hat{\boldsymbol{h}}$, first run $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y)$. Pick $\boldsymbol{s} = (s, s_1, \ldots, s_{w_2}), \hat{\boldsymbol{s}} \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$ Output the ciphertext as $\mathsf{CT} = (\boldsymbol{C}, C_0)$:
$$\boldsymbol{C} = g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})} g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})} \in \mathbb{G}^{w_1}, \qquad C_0 = (e(g_1, g_1)^\alpha)^s M \in \mathbb{G}_T. \tag{3}$$

- $\mathsf{SFKeyGen}(X, \mathsf{MSK}, \mathsf{PK}, g_2, \mathsf{type}, \hat{\alpha}, \hat{\boldsymbol{h}})$: Upon inputs $X, \mathsf{MSK}, \mathsf{PK}, g_2$ and $\mathsf{type} \in \{1, 2, 3\}, \hat{\alpha} \in \mathbb{Z}_N$, first run $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X)$. Pick $\boldsymbol{r}, \hat{\boldsymbol{r}} \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$. Output the secret key SK:
$$\boldsymbol{K} = \begin{cases} g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \cdot \boldsymbol{R}_3 & \text{if } \mathsf{type} = 1 \qquad (4) \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \cdot \boldsymbol{R}_3 & \text{if } \mathsf{type} = 2 \qquad (5) \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\boldsymbol{0},\boldsymbol{0})} \cdot \boldsymbol{R}_3 & \text{if } \mathsf{type} = 3 \qquad (6) \end{cases}$$

  Note that in computing type-1 (resp., type-3) semi-functional keys, $\hat{\alpha}$ (resp., $\hat{\boldsymbol{h}}$) is not needed.

4.4 Security Theorems for Our Framework

We obtain two security theorems for the generic construction. The first one is for the case when the underlying pair encoding scheme is doubly selectively master-key hiding, and hence we achieve tighter reduction of $O(q_1)$. The second one is for the case when the underlying pair encoding scheme is perfectly master-key hiding, and hence only normal reduction cost of $O(q_{\text{all}})$ is achieved.

Theorem 1. *Suppose that a pair encoding scheme* P *for predicate* R *is selectively and co-selectively master-key hiding in* $\mathcal{G}$*, and the Subgroup Decision Assumption 1,2,3 hold in* $\mathcal{G}$*. Also, suppose that* R *is domain-transferable. Then the construction* $\mathsf{FE}(\mathsf{P})$ *in* $\mathcal{G}$ *of function encryption for predicate* R *is fully secure. More precisely, for any PPT adversary* $\mathcal{A}$*, there exist PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5$*, whose running times are essentially the same as* $\mathcal{A}$*, such that for any* λ*,*
$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SD1}}(\lambda) + (2q_1 + 3)\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{SD2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SD3}}(\lambda) + q_1\mathsf{Adv}_{\mathcal{B}_4}^{\mathsf{CMH}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_5}^{\mathsf{SMH}}(\lambda),$$
where q_1 *is the number of queries in phase 1.*

Figure 1: The sequence of games in the security proof.

G_{res} :The restriction becomes $R_{p_2}(X_j, Y^\star) = 0$. (Instead of $R_N(X_j, Y^\star) = 0$).

G_0 :Modify $^{(1)}$ $\boxed{\mathsf{SFsetup}(1^\lambda, \kappa) \to (\mathsf{PK}, \mathsf{MSK}, g_2, \hat{\boldsymbol{h}})}$ (in the Setup phase).

Modify $^{(3)}$ $\boxed{\mathsf{CT}^\star \leftarrow \mathsf{SFEncrypt}(Y, M_b, \mathsf{PK}, g_2, \hat{\boldsymbol{h}})}$.

$\mathsf{G}_{k,1}$:Modify $^{(2)}$ $\boxed{\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N,\ \mathsf{SK}_j \leftarrow \begin{cases} \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 3, \hat{\alpha}_j, \boldsymbol{0}) & \text{if } j < k \\ \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 1, 0, \hat{\boldsymbol{h}}) & \text{if } j = k \\ \quad \mathsf{KeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}) & \text{if } j > k \end{cases}}$ (in Phase 1).

$\mathsf{G}_{k,2}$:Modify $^{(2)}$ $\boxed{\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N,\ \mathsf{SK}_j \leftarrow \begin{cases} \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 3, \hat{\alpha}_j, \boldsymbol{0}) & \text{if } j < k \\ \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 2, \hat{\alpha}_j, \hat{\boldsymbol{h}}) & \text{if } j = k \\ \quad \mathsf{KeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}) & \text{if } j > k \end{cases}}$ (in Phase 1).

$\mathsf{G}_{k,3}$:Modify $^{(2)}$ $\boxed{\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N,\ \mathsf{SK}_j \leftarrow \begin{cases} \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 3, \hat{\alpha}_j, \boldsymbol{0}) & \text{if } j \le k \\ \quad \mathsf{KeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}) & \text{if } j > k \end{cases}}$ (in Phase 1).

G_{q_1+1}:Modify $^{(4)}$ $\boxed{\mathsf{SK}_j \leftarrow \quad \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 1, 0, \hat{\boldsymbol{h}})}$ (in Phase 2).

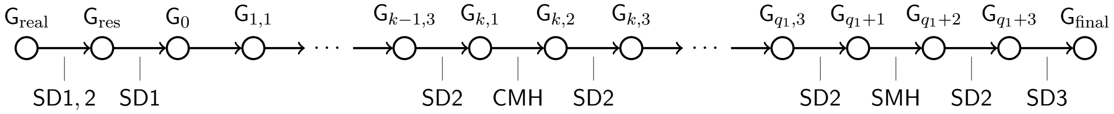
G_{q_1+2}:Insert $\boxed{\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N}$ at the begin of Phase 2.

Modify $^{(4)}$ $\boxed{\mathsf{SK}_j \leftarrow \quad \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 2, \hat{\alpha}, \hat{\boldsymbol{h}})}$ (in Phase 2).

G_{q_1+3}:Modify $^{(4)}$ $\boxed{\mathsf{SK}_j \leftarrow \quad \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 3, \hat{\alpha}, \boldsymbol{0})}$ (in Phase 2).

$\mathsf{G}_{\text{final}}$:Modify $^{(3)}$ $\boxed{M \xleftarrow{\$} \mathcal{M},\quad \mathsf{CT}^\star \leftarrow \quad \mathsf{SFEncrypt}(Y, M, \mathsf{PK}, g_2, \hat{\boldsymbol{h}})}$.

Security Proof Structure for Theorem 1. We use a sequence of games in the following order:



where each game is defined as follows. G_{real} is the actual security game, and each of the following game is defined exactly as *its previous game* in the sequence except the specified modification that is defined in Fig. 1. For notational purpose, let $\mathsf{G}_{0,3} := \mathsf{G}_0$. In the diagram, we also write the underlying assumptions used for indistinguishability between adjacent games. The proofs of indistinguishability between all these adjacent games are given in §A.1.

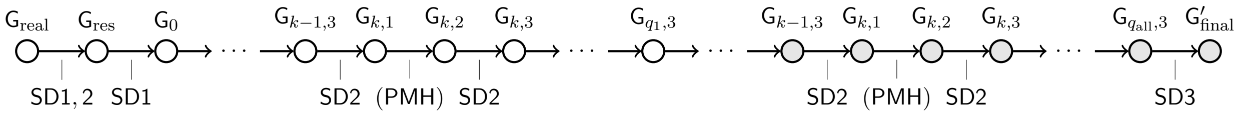
We also obtain the theorem for the case where the encoding is perfectly master-key hiding.

Theorem 2. *Suppose that a pair encoding scheme* P *for predicate* R *is perfectly master-key hiding, and the Subgroup Decision Assumption 1,2,3 hold in* $\mathcal{G}$*. Suppose also that* R *is domain-transferable. Then the construction* $\mathsf{FE}(\mathsf{P})$ *in* $\mathcal{G}$ *of function encryption for predicate* R *is fully secure. More precisely, for any PPT adversary* $\mathcal{A}$*, there exist PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$*, whose running times are essentially the same as* $\mathcal{A}$*, such that for any* λ*,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) \le 2\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{SD1}}(\lambda) + (2q_{\text{all}} + 1)\mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{SD2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SD3}}(\lambda).$$

where $q_{\text{all}} = q_1 + q_2$ *denotes the number of all queries.*

Security Proof Structure for Theorem 2. We use a sequence of games in the following order:



where each game is defined as in the proof of Theorem 1, with the following exceptions. The additional games that are not in the proof of Theorem 1 are indicated in the gray color. The new games $\mathsf{G}_{k,1}, \mathsf{G}_{k,2}, \mathsf{G}_{k,3}$ for $q_1 < k \leq q_{\text{all}}$ are defined as those for $1 \leq k \leq q_1$ in Fig. 1 except that the modification from its previous game is done for box (4) in Phase 2, instead of box (1) in Phase 1. The new game $\mathsf{G}'_{\text{final}}$ has the same modification as that of $\mathsf{G}_{\text{final}}$ in Fig. 1, but now it is modified from $\mathsf{G}_{q_{\text{all}},3}$, and hence $\hat{\alpha}_j$ is varied for all keys. All the lemmata are the same as the proof of Theorem 1 except only between $\mathsf{G}_{k,1}$ from $\mathsf{G}_{k,2}$ where we will use perfect master-key hiding security instead. This indistinguishability is shown in §A.2.

5 Efficient Fully Secure IBE with Tighter Reduction

We first construct an encoding scheme for the simplest predicate, namely the equality relation, and hence obtain a new IBE scheme. This is shown as Scheme 1. It is similar to the Boneh-Boyen IBE [4] (and Lewko-Waters IBE [23]), with the exception that we have one more element in each of ciphertext and key. Their roles will be explained below. The encoding scheme can be proved perfectly master-key hiding due to the fact that $f(x) = h_1 + h_2 x$ is pairwise independent function (this is also used in [23]). The novelty is that we can prove the SMH security (with tight reduction to 3DH). Note that the CMH security is implied by perfect master-key hiding. Hence, from Theorem 1, we obtain a fully secure IBE with $O(q_1)$ reduction to SD2, plus tight reduction to 3DH, SD1, SD3. (See Theorem 4 below). [5]

Pair Encoding Scheme 1: IBE with Tighter Reduction
Param $\to 3$. Denote $\boldsymbol{h} = (h_1, h_2, h_3)$.
Enc1$(X) \to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = (\alpha + r(h_1 + h_2 X) + u h_3, \quad r, \quad u)$ where $\boldsymbol{r} = (r, u)$.
Enc2$(Y) \to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = (s, \quad s(h_1 + h_2 Y), \quad s h_3)$ where $\boldsymbol{s} = s$.

Lemma 3. *Scheme 1 is selectively master-key hiding under 3DH with tight reduction.*

Proof. Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the SMH game against Scheme 1. We construct a simulator $\mathcal{B}$ that solves 3DH. $\mathcal{B}$ takes as an input the 3DH challenge, $\boldsymbol{D} = (g_2, g_2^a, g_2^b, g_2^z, g_1, g_3)$ and $T = g_2^{\tau + abz}$, where either $\tau = 0$ or $\tau \xleftarrow{\$} \mathbb{Z}_{p_2}$. $\mathcal{B}$ first gives (g_1, g_2, g_3) to $\mathcal{A}$.

Ciphertext Query (to $\mathcal{O}^1$). The game begins with $\mathcal{A}$ making a query for identity $Y^\star$ to $\mathcal{O}^1$. $\mathcal{B}$ picks $h'_1, h'_2, h'_3 \xleftarrow{\$} \mathbb{Z}_N$ and defines $\boldsymbol{h} = (h_1, h_2, h_3)$ by *implicitly* setting $g_2^{h_1} = {g_2}^{h'_1} {g_2}^{-Y^\star za}$, $g_2^{h_2} = {g_2}^{h'_2} g_2^{za}$, $g_2^{h_3} = {g_2}^{h'_3} g_2^{z}$. Note that only the last term is computable. $\mathcal{B}$ picks $s \xleftarrow{\$} \mathbb{Z}_N$ and computes $\boldsymbol{C} = {g_2}^{\boldsymbol{c}(s,\boldsymbol{h})} = (C_1, C_2, C_3)$ as:

$$C_1 = g_2^s, \qquad C_2 = g_2^{s(h'_1 + h'_2 Y^\star)}, \qquad C_3 = (g_2^{h_3})^s.$$

[5] Compared to the recent IBE of [8], their scheme has the reduction cost that does not depend on the number of queries; they achieved $O(\ell)$ reduction to DLIN, while the public key size is $O(\ell)$, where ℓ is the identity length. Ours has $O(1)$ public key size.

Obviously, C_1, C_3 are properly distributed. C_2 is properly distributed due to the cancellation of unknown za in the exponent: $h_1 + h_2 Y^\star = (h'_1 - Y^\star za) + (h'_2 + za)Y^\star = h'_1 + h'_2 Y^\star$.

Key Query (to $\mathcal{O}^2$). When $\mathcal{A}$ makes the j-th key query for $X_j (\neq Y^\star)$, $\mathcal{B}$ first computes a temporary key $\boldsymbol{K}' = (K'_1, K'_2, K'_3)$ where

$$K'_1 = T\Big((g_2^b)^{\frac{1}{X_j - Y^\star}}\Big)^{h'_1 + h'_2 X_j}, \qquad K'_2 = (g_2^b)^{\frac{1}{X_j - Y^\star}}, \qquad K'_3 = 1.$$

We then claim that $\boldsymbol{K}' = {g_2}^{\boldsymbol{k}_{X_j}(\tau, \boldsymbol{r}'_j, \boldsymbol{h})}$, where $\boldsymbol{r}'_j = (r'_j, u'_j) = (\frac{b}{X_j - Y^\star}, 0)$. This holds since $K'_1 = {g_2}^{(\tau + abz) + (\frac{b}{X_j - Y^\star})(h'_1 + h'_2 X_j)} = {g_2}^{\tau + (\frac{b}{X_j - Y^\star})((h'_1 - Y^\star za) + (h'_2 + za)X_j)} = {g_2}^{\tau + r'_j(h_1 + h_2 X_j)}$, where the unknown element abz in the exponent term $r'_j(h_1 + h_2 X_j)$ is simulated by using abz from T. A crucial point here is that $\boldsymbol{K}'$ is *not yet* properly distributed as r'_j is not independent among j (since all r'_j are determined from b). We re-randomize it by picking $r''_j, u''_j \xleftarrow{\$} \mathbb{Z}_N$ and computing

$$K_1 = K'_1 \Big({g_2}^{r''_j}\Big)^{h'_1 + h'_2 X_j} (g_2^z)^{u''_j}, \qquad K_2 = K'_2 {g_2}^{r''_j}, \qquad K_3 = K'_3 {g_2}^{u''_j} (g_2^a)^{-r''_j (X_j - Y^\star)}.$$

This is a properly distributed $\boldsymbol{K} = {g_2}^{\boldsymbol{k}_{X_j}(\tau, \boldsymbol{r}_j, \boldsymbol{h})}$ with $\boldsymbol{r}_j = (r_j, u_j) = (\frac{b}{X_j - Y^\star} + r''_j, u''_j - a r''_j (X_j - Y^\star))$.

Guess. The algorithm $\mathcal{B}$ has properly simulated $\boldsymbol{K} = {g_2}^{\boldsymbol{k}_{X_j}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})}$ with $\alpha = 0$ if $\tau = 0$, and α is random if τ is random (since $\alpha = \tau$). $\mathcal{B}$ thus outputs the corresponding guess from $\mathcal{A}$. The advantage of $\mathcal{B}$ is thus equal to that of $\mathcal{A}$. □

Remark 4 (Randomizer Technique)**.** Our proof much resembles the Boneh-Boyen technique [4], with a crucial exception that here we need to establish the indistinguishability in $\mathbb{G}$ (for our purpose of master-key hiding notion), instead of $\mathbb{G}_T$ (for the purpose of proving security for BB-IBE). Therefore, intuitively, instead of embedding only g^a to the parameter $g^{\boldsymbol{h}}$ as usual, we need to embed g^{az} so as to obtain the target element g^{abz} in $\mathbb{G}$ when combining with r (which uses b). This is in contrast to BB-IBE, where the target $e(g,g)^{abz}$ is in $\mathbb{G}_T$. Now that $g^{\boldsymbol{h}}$ contains non-trivial term g^{az}, we cannot re-randomize r in keys. To solve this, we introduce u as a "randomizer" via g^a. This is why we need one more element than BB-IBE. This technique is implicit in ABE of [26]. We will use this technique also for other instantiations throughout the paper.

From Lemma 3 and Theorem 1, we thus obtain the following theorem for full security of our IBE.

Theorem 4. *Suppose that the 3DH and the Subgroup Decision Assumption 1,2,3 hold in $\mathcal{G}$. Then the construction* FE(Scheme 1) *in $\mathcal{G}$ of IBE is fully secure. More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$, whose running times are essentially the same as $\mathcal{A}$, such that for any λ,*

$$\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}_1}(\lambda) + (2q_1 + 3)\mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{SD3}}_{\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{\mathsf{3DH}}_{\mathcal{B}_4}(\lambda),$$

where q_1 is the number of queries in phase 1.

6 Fully Secure FE for Regular Languages

Predicate Definition of FE for Regular Languages. In functional encryption for regular languages, we have a key associated to the description of a deterministic finite automata (DFA) M, while a ciphertext is associated to a string w, and $R(M, w) = 1$ if the automata M accepts the string w. A DFA M is a 5-tuple $(Q, \Lambda, \mathcal{T}, q_0, F)$ in which Q is the set of states $Q = \{q_0, q_1, \ldots, q_{n-1}\}$, Λ is the alphabet set, $\mathcal{T}$ is the set of transitions, in which each transition is of the form $(q_x, q_y, \sigma) \in Q \times Q \times \Lambda$, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of accepted states. We say that M accepts a string $w = (w_1, w_2, \ldots, w_\ell) \in \Lambda^*$ if there exists a sequence of states $\rho_0, \rho_1, \ldots, \rho_\ell \in Q$ such that $\rho_0 = q_0$, for $i = 1$ to ℓ we have $(\rho_{i-1}, \rho_i, w_i) \in \mathcal{T}$, and $\rho_\ell \in F$. The primitive is also called DFA-based FE. This primitive is important since it has a unique unbounded feature that one key for machine M can operate on input string w of *arbitrary* sizes.

Such an FE was proposed by Waters [38], where selective security was achieved. No fully secure realization is known so far. Waters also suggested that classical dual system techniques could be used, but only with the restricted version of the primitive where some bounds must be posed. This is clearly not satisfactory since the bound would negate the motivation of having arbitrary string sizes for the ciphertext attribute. [6]

6.1 Previous FE for Regular Languages by Waters

We first review the Waters' scheme [38] by extracting the underlying pair encoding into Scheme 2.

Pair Encoding Scheme 2: Waters' FE for Regular Languages

$\mathsf{Param}(|\Lambda|) \to |\Lambda| + 3$. Denote $\boldsymbol{h} = (z, h_{\text{start}}, h_{\text{end}}, (h_\sigma)_{\sigma \in \Lambda})$.

$\mathsf{Enc1}(M)$ For any DFA $M = (Q, \Lambda, \mathcal{T}, q_0, F)$, let $n = |Q|$, let $m = |\mathcal{T}|$, and parse $\mathcal{T} = \{\, (q_{x_t}, q_{y_t}, \sigma_t) \mid t \in [1, m] \,\}$.

$$\to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = \big(k_1, k_2, \{k_{3,t}, k_{4,t}, k_{5,t}\}_{t \in [1,m]}, \{k_{6,x}, k_{7,x}\}_{x \in F}\big) :$$

$$\left\{ \begin{array}{lll} k_1 = d_0 + h_{\text{start}} r_{\text{start}}, & k_2 = r_{\text{start}}, & \\ k_{3,t} = r_t, & k_{4,t} = -d_{x_t} + z r_t, & k_{5,t} = d_{y_t} + h_{\sigma_t} r_t, \\ k_{6,x} = -\alpha + d_x + h_{\text{end}} r_{\text{end},x}, & k_{7,x} = r_{\text{end},x}, & \end{array} \right\}$$

where $\boldsymbol{r} = (r_{\text{start}}, r_1, \ldots, r_m, \{d_x\}_{q_x \in Q}, \{r_{\text{end},x}\}_{x \in F})$.

$\mathsf{Enc2}(w)$ For $w \in (\mathbb{Z}_N)^*$, let $\ell = |w|$, and parse $w = (w_1, \ldots, w_\ell)$.

$$\to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = \big(c_1, \{c_{2,i}\}_{i \in [0,\ell]}, \{c_{3,i}\}_{i \in [1,\ell]}, c_4\big) :$$

$$\big\{ c_1 = s_0 h_{\text{start}}, \quad c_{2,i} = s_i, \quad c_{3,i} = s_{i-1} z + s_i h_{w_i}, \quad c_4 = s_\ell h_{\text{end}}, \big\}$$

where $\boldsymbol{s} = (s_\ell, s_0, s_1, \ldots, s_{\ell-1})$. (Hence $s = s_\ell$).

Correctness. The correctness can be shown as follows. If $R(M, w) = 1$, we have that there is a sequence of states $\rho_0, \rho_1, \ldots, \rho_\ell \in Q$ such that $\rho_0 = q_0$, for $i = 1$ to ℓ we have $t_i = (\rho_{i-1}, \rho_i, w_i) \in \mathcal{T}$, and $\rho_\ell \in F$. Let $(q_{x_{t_i}}, q_{y_{t_i}}, \sigma_{t_i}) = (\rho_{i-1}, \rho_i, w_i)$. Hence, we have the following linear combination of $k_\iota c_j$ terms: $k_1 c_{2,0} - k_2 c_1 + \sum_{i \in [1,\ell]} (k_{4,t_i} c_{2,i-1} - k_{3,t_i} c_{3,i} + k_{5,t_i} c_{2,i}) - k_{6,y_{t_\ell}} c_{2,\ell} + k_{7,y_{t_\ell}} c_4 = \alpha s$.

Lemma 5. *Scheme 2 is not perfectly master-key hiding.*

Proof. We consider M, w such that M, upon input w, transits the initial state through a state $q \in F$ after the j-th symbol of w but eventually the last state is not in F, hence $R(M, w) = 0$. More precisely, we have a sequence $\rho_0, \rho_1, \ldots, \rho_j \in Q$ such that $\rho_0 = q_0$, for $i = 1$ to j we have $t_i = (\rho_{i-1}, \rho_i, w_i) \in \mathcal{T}$, and $\rho_j = q \in F$. Now we show how to compute α from $\boldsymbol{k}_M(\alpha, \boldsymbol{r}, \boldsymbol{h})$ and $\boldsymbol{c}_w(\boldsymbol{s}, \boldsymbol{h})$.

Firstly we claim that from our setting of w, we can compute $s_j d_q$. Intuitively, this is done by mimicking the decryption mechanism but only until this j-th symbol, instead of the ℓ-th symbol. More precisely, we have $k_1 c_{2,0} - k_2 c_1 + \sum_{i \in [1,j]} (k_{4,t_i} c_{2,i-1} - k_{3,t_i} c_{3,i} + k_{5,t_i} c_{2,i}) = s_j d_q$.

[6] A recent work [35] proposed a candidate for such a bounded scheme. Since classical dual system techniques are used there, it is expected that its underlying pair encoding would be perfectly master-key hiding.

Now from $s_j d_q$ and the known term s_j, the term d_q is exposed. From known terms $s_\ell, s_\ell h_{\mathrm{end}}$, we know h_{end}. From this and a known term $r_{\mathrm{end},q}$ and d_q, we can extract α from $-\alpha+d_q+h_{\mathrm{end}}r_{\mathrm{end},q}$. □

6.2 Our Fully Secure FE for Regular Languages

Simplification. We first observe that it is simpler and without loss of generality if we consider machines where $|F| = 1$, *i.e.,* it has only one accepted state. To see this, we show how to map any $(M, \boldsymbol{w})$ to $(M', \boldsymbol{w}')$ where M' has one accepted state and claim that $R(M', \boldsymbol{w}') = 1$ if and only if $R(M, \boldsymbol{w}) = 1$. From M, we construct M' by adding a new state q^* and new transitions (q, q^*, σ^*) for all $q \in F$ with a new fixed special symbol σ^*. We then map $\boldsymbol{w}$ to $(\boldsymbol{w}, \sigma^*)$, *i.e.,* concatenating σ^* with $\boldsymbol{w}$. The claim then follows straightforwardly.

Motivation for Large Universe. Waters' scheme operates over *small-universe* alphabet sets, *i.e.,* $|\Lambda|$ is of polynomial size. We argue that this small-universe nature makes the system less efficient than other less-advanced FE for the same functionality. For example, we consider IBE, of which predicate determines equality over two identity $X, Y \in \{0,1\}^n$. To construct DFA that operates over small-size universe to determine if $X = Y$ would require $\Theta(\log n)$ transition, which is not so satisfactory for such a simple primitive.

Our Encoding Scheme. We propose a new scheme which is *fully secure* and operates over *large-universe* alphabet sets, *i.e.,* $|\Lambda|$ is of super-polynomial size, namely we use $\Lambda = \mathbb{Z}_N$. This is also called *unbounded* alphabet universe, since the parameter size will not depend on the alphabet universe. (That is, the resulting FE scheme has a constant-size public key). Our encoding scheme is shown as Scheme 3.

Pair Encoding Scheme 3: Our FE for Regular Languages

Param $\to 8$. Denote $\boldsymbol{h} = (h_0, h_1, h_2, h_3, h_4, \phi_1, \phi_2, \eta)$.

For any DFA $M = (Q, \mathbb{Z}_N, \mathcal{T}, q_0, q_{n-1})$, where $n = |Q|$,
let $m = |\mathcal{T}|$, and parse $\mathcal{T} = \{(q_{x_t}, q_{y_t}, \sigma_t) | t \in [1, m]\}$.

Enc1(M) $\to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = (k_1, k_2, k_3, k_4, k_5, \{k_{6,t}, k_{7,t}, k_{8,t}\}_{t\in[1,m]})$:

$$\left\{\begin{array}{lll} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_4 = r_0, & k_5 = -u_0 + r_0 h_0, & k_{6,t} = r_t, \\ k_{7,t} = u_{x_t} + r_t(h_1 + h_2\sigma_t), & k_{8,t} = -u_{y_t} + r_t(h_3 + h_4\sigma_t) & \end{array}\right\}$$

where $u_{n-1} := \phi_2 r$ and $\boldsymbol{r} = (r, u, r_0, r_1, \ldots, r_m, \{u_x\}_{q_x \in Q \smallsetminus \{q_{n-1}\}})$.

For $w \in (\mathbb{Z}_N)^*$, let $\ell = |w|$, and parse $w = (w_1, \ldots, w_\ell)$.

Enc2(w) $\to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = (c_1, c_2, c_3, c_4, \{c_{5,i}\}_{i\in[0,\ell]}, \{c_{6,i}\}_{i\in[1,\ell]})$:

$$\left\{\begin{array}{lll} c_1 = s, & c_2 = s\eta, & c_3 = -s\phi_1 + s_\ell\phi_2, \\ c_4 = s_0 h_0, & c_{5,i} = s_i, & c_{6,i} = s_{i-1}(h_1 + h_2 w_i) + s_i(h_3 + h_4 w_i) \end{array}\right\}$$

where $\boldsymbol{s} = (s, s_0, s_1, \ldots, s_\ell)$.

Correctness. The correctness can be shown by providing linear combination of $k_\iota c_j$ which summed up to αs. When $R(M, w) = 1$, we have that there is a sequence of states $\rho_0, \rho_1, \ldots, \rho_\ell \in Q$ such that $\rho_0 = q_0$, for $i = 1$ to ℓ we have $(\rho_{i-1}, \rho_i, w_i) \in \mathcal{T}$, and $\rho_\ell \in F$. Let $(q_{x_{t_i}}, q_{y_{t_i}}, \sigma_{t_i}) = (\rho_{i-1}, \rho_i, w_i)$. Therefore, we have the following bilinear combination:

$$k_1c_1 - k_2c_2 + k_3c_3 - k_4c_4 + k_5c_{5,0} + \sum_{i\in[1,\ell]} (-k_{6,t_i}c_{6,i} + k_{7,t_i}c_{5,i-1} + k_{8,t_i}c_{5,i}) = \alpha s.$$

This holds since for any $i \in [1, \ell]$, we have $-k_{6,t_i} c_{6,i} + k_{7,t_i} c_{5,i-1} + k_{8,t_i} c_{5,i} = s_{i-1} u_{x_{t_i}} - s_i u_{y_{t_i}}$. The sum of these terms for all $i \in [1, \ell]$ will form chaining cancelations and results in $s_0 u_{x_{t_1}} - s_\ell u_{y_{t_\ell}} = s_0 u_0 - s_\ell u_{n-1} = s_0 u_0 - s_\ell \phi_2 r$. Adding this to the rest, we obtain αs.

Intuition. Our construction is built upon that of Waters' (Scheme 2). There are roughly five differences to ours. First, for our purpose of unbounded alphabet, we represent an alphabet by a function $f(\sigma) = h_1 + h_2(\sigma)$, instead of preparing one parameter h_σ per alphabet. Second, and perhaps most importantly, for our *co-selective* master-key hiding proof, our scheme uses a "balance" approach: both the terms that are related to transitions, which are $k_{7,t}, k_{8,t}$, are functions of the corresponding alphabet σ_t. This is in contrast with Waters' scheme where only one of the transition terms is a function of σ_t (the term $k_{5,t}$ in Scheme 2). The reason for our scheme to require this is to perform a certain type of cancellation in the proof for the co-selective security (more precisely, the last case for ciphertext cancelation in the proof in §B.3). Third, we structure the terms in two different layers and link both via c_3. This technique is used for proving primitives with large universes, implicitly used in [29]. Fourth, the term k_2 is prepared for the randomizer technique, as stated in Remark 4. Finally, our scheme uses the wlog condition that $|F| = 1$ (see Simplification above), hence the term regarding h_{end} from Waters' scheme disappears.

Lemma 6. *Scheme 3 is not perfectly master-key hiding.* (The proof is in §B.1).

We now state the doubly selective security of our encoding. We introduce two new assumptions. The first one is for proving selective master-key hiding security, and hence is similar to the assumption for for proving selective security of the Waters' scheme in [38]. The second one is for proving co-selective master-key hiding security, where we will use completely new techniques. The generic security of these assumptions will be shown in §E.

Definition 3 (ℓ-$\mathsf{EDHE1}$ Assumption)**.** The ℓ-Expanded Diffie-Hellman Exponent Assumption-1 in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, b, c, d_1, \ldots, d_{\ell+1}, f, z \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$ (for simplicity). Suppose that an adversary is given a target element $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of

$$\begin{aligned}
& g, g^a, g^b, g^{a/f}, g^{1/f}, g^{a^\ell c/z} \\
\forall_{i\in[1,\ell+1]} \quad & g^{a^i/d_i}, g^{a^i bf} \\
\forall_{i\in[0,\ell]} \quad & g^{a^i c}, g^{bd_i}, g^{bd_i/f}, g^{abd_i/f} \\
\forall_{i\in[1,2\ell+1], i\neq\ell+1,\ j\in[1,\ell+1]} \quad & g^{a^i c/d_j} \\
\forall_{i\in[2,2\ell+2],\ j\in[1,\ell+1]} \quad & g^{a^i bf/d_j} \\
\forall_{i,j\in[1,\ell+1], i\neq j} \quad & g^{a^i bd_j/d_i},
\end{aligned}$$

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

Lemma 7. *Scheme 3 is selectively master-key hiding under the ℓ-$\mathsf{EDHE1}$ assumption with tight reduction, where ℓ is the length of the one-ciphertext query $w^\star$.* (The proof is in §B.2).

Definition 4 ((n,m)-$\mathsf{EDHE2}$ Assumption)**.** The (n,m)-Expanded Diffie-Hellman Exponent assumption-2 in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, b, c, d_1, \ldots, d_m, z \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$ (for simplicity). Suppose that an adversary is given a target element $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of

$$\begin{array}{rl}
 & g, g^a, g^b, g^{a^{n-1}c/z} \\
\forall_{i\in[1,n],\ j,j'\in[1,m],j\neq j'} & g^{a^i/d_j^2}, g^{a^ib/d_j}, g^{d_j}, g^{a^id_j/d_{j'}^2}, g^{a^ibd_j/d_{j'}}, g^{a^i/d_j^6}, g^{a^id_j/d_{j'}^6}, \\
\forall_{i\in[0,n-1]} & g^{a^ic}, g^{a^ibcd_j}, \\
\forall_{i\in[0,n],\ j\in[1,m]} & g^{a^ibcd_j^5}, \\
\forall_{i\in[1,2n-1],\ j,j'\in[1,m],j\neq j'} & g^{a^ibcd_j/d_{j'}^2}, g^{a^ibcd_j^5/d_{j'}^6}, \\
\forall_{i\in[1,2n-1],i\neq n\ j\in[1,m]} & g^{a^ibc/d_j}, \\
\forall_{i\in[1,2n-1],\ j,j'\in[1,m]} & g^{a^ic/d_j^2}, g^{a^ib^2cd_j/d_{j'}}, g^{a^ibcd_j/d_{j'}^6}, g^{a^ic/d_j^6}, g^{a^ibcd_j^5/d_{j'}^2}, g^{a^ib^2cd_j^5/d_{j'}}
\end{array}$$

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

Lemma 8. *Scheme 3 is co-selectively master-key hiding under the (n,m)-EDHE2 assumption with tight reduction, where $n = |Q|$ is the number of states, $m = |\mathcal{T}|$ is the number of transitions of the one-key query DFA $M^\star$.* (The proof is in §B.3).

7 Fully Secure ABE with Short Ciphertexts and Unbounded ABE

In this section, we present our two main ABE instantiations, namely fully secure ABE with short ciphertexts and fully secure unbounded ABE with large-universe. We first show their scheme descriptions. We then argue that they are special cases of our new primitive called Key-Policy Doubly Spatial Encryption Scheme (KP-DSE), which we introduce in the following subsection. The security theorems for both instantiations then follow from that of KP-DSE.

Predicate Definition of ABE. We first review the definition for ABE for boolean formulae. Let $\mathcal{U}$ be a universe of attributes. In Key-Policy ABE, a key is associated to a policy, which is described by a boolean formulae Ψ over $\mathcal{U}$, while a ciphertext is associated to an attribute set $S \subseteq \mathcal{U}$. We have $R(\Psi, S) = 1$ if the evaluation of Ψ returns `true` when setting attributes in S as `true` and the others (in Ψ) as `false`.

ABE with *large-universe* is a variant where $\mathcal{U}$ is of super-polynomial size. *Unbounded* ABE is a variant where there is no restriction on any sizes of policies Ψ, attribute sets S, or the maximum number of attribute repetition in a policy. In a *bounded* ABE scheme, the corresponding bounds (*e.g.,* the maximum size of S) will be described as indexes inside κ for the predicate family.

A boolean formula can be equivalently described by a linear secret sharing (LSS) scheme (A, π) over $\mathbb{Z}_N$, where A is a matrix in $\mathbb{Z}_N^{m\times k}$ and $\pi : [1,m] \to \mathcal{U}$, for some m, k. We briefly review the definition of LSS. Consider a set $S \subseteq \mathcal{U}$. Let A_S be the sub-matrix of A that takes exactly all the rows in $\pi(S) = \{\, \pi(j) \mid j \in S \,\}$. We say that (A,π) accepts S if and only if $(1, \mathbf{0}) \in \mathsf{RowSp}(A_S)$. An LSS scheme consists of two algorithms. First, Share takes as input $s \in \mathbb{Z}_N$ (a secret to be shared), and chooses $v_2, \ldots, v_k \xleftarrow{\$} \mathbb{Z}_N$, sets $\boldsymbol{v} = (s, v_2 \ldots, v_k)$, and outputs $A_i\boldsymbol{v}^\top$ as the i-th share, where A_i is the i-th row of A, for $i \in [1,m]$. Second, $\mathsf{Reconstruct}$ takes as input S such that (A,π) accepts S, and outputs a set of constants $\{\mu_i\}_{i\in I}$, where $I := \{\, i \mid \pi(i) \in S \,\}$, which has a reconstruction property: $\sum_{i\in I} \mu_i(A_i\boldsymbol{v}^\top) = s$. Such a set of constants can be computed since (A,π) accepts S, we have $(1, \mathbf{0}) \in \mathsf{RowSp}(A_S)$, and hence we choose $\{\mu_i\}_{i\in I}$ with $\sum_{j\in S} \mu_j A_j = (1, \mathbf{0})$.

7.1 Fully-Secure Unbounded ABE with Large Universes

Our pair encoding scheme for unbounded KP-ABE with large universes is shown as Scheme 4. We can see that the parameter size is constant, and we can deal with any sizes of attribute policies, attribute sets, while the attribute universe is $\mathbb{Z}_N$. The structure of our scheme is similar to the selectively secure ABE of [29].

Pair Encoding Scheme 4: Unbounded KP-ABE with Large Universes	
Param	$\to 6$. Denote $\boldsymbol{h} = (h_0, h_1, \phi_1, \phi_2, \phi_3, \eta)$.
$\mathsf{Enc1}(A, \pi)$	For LSS $A \in \mathbb{Z}_N^{m\times k}$, $\pi : [1, m] \to \mathbb{Z}_N$ (π needed not be injective). $\to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = \big(k_1, k_2, k_3, \{k_{4,i}, k_{5,i}, k_{6,i}\}_{i\in[1,m]}\big)$: $\left\{\begin{array}{lll} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_{4,i} = A_i\boldsymbol{v}^\top + r_i\phi_3, & k_{5,i} = r_i, & k_{6,i} = r_i(h_0 + h_1\pi(i)) \end{array}\right\}$ where $v_1 = r\phi_2$, $\boldsymbol{r} = (r, u, r_1, \ldots, r_m, v_2, \ldots, v_k)$, $\boldsymbol{v} = (v_1, \ldots, v_k)$.
$\mathsf{Enc2}(S)$	For $S \subseteq \mathbb{Z}_N$. $\to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = \big(c_1, c_2, c_3, c_4, \{c_{5,y}, c_{6,y}\}_{y\in S}\big)$: $\left\{\begin{array}{lll} c_1 = s, & c_2 = s\eta, & c_3 = s\phi_1 + w\phi_2, \\ c_4 = w, & c_{5,y} = w\phi_3 + s_y(h_0 + h_1 y), & c_{6,y} = s_y \end{array}\right\}$ where $\boldsymbol{s} = (s, w, \{s_y\}_{y\in S})$.

Correctness. When $R((A, \pi), S) = 1$, let $I = \{\, i \in [1, m] \mid \pi(i) \in S \,\}$, we have reconstruction coefficients $\{\mu_i\}_{i\in I}$ such that $\sum_{i\in I} \mu_i A_i\boldsymbol{v}^\top = v_1 = r\phi_2$. Therefore, we have the following linear combination of the $k_\iota c_j$ terms:

$$k_1c_1 - k_2c_2 - k_3c_3 + \sum_{i\in I} \mu_i\big(k_{4,i}c_4 - k_{5,i}c_{5,\pi(i)} + k_{6,i}c_{6,\pi(i)}\big) = \alpha s - rw\phi_2 + \sum_{i\in I} \mu_i(A_i\boldsymbol{v}^\top w) = \alpha s.$$

We show its doubly selective security in Corollary 13 and 14 below. Now we show that it is not perfectly master-key hiding. The underlying encoding scheme of [29] is also not perfectly master-key hiding by similar argument.

Lemma 9. *Scheme 4 is not perfectly master-key hiding.*

Proof. We consider S and (A, π) such that there exists j where $\pi(j) \in S$ but $R((A, \pi), S) = 0$. Now we will show how to compute α from $\boldsymbol{k}_{(A,\pi)}(\alpha, \boldsymbol{r}, \boldsymbol{h})$ and $\boldsymbol{c}_S(\boldsymbol{s}, \boldsymbol{h})$. Firstly from $k_{5,j} = r_j$ and $k_{6,j} = r_j(h_0 + h_1\pi(j))$, we know $h_0 + h_1\pi(j)$. Now since we know $c_{6,\pi(j)} = s_{\pi(j)}$ and $c_4 = w$, the term ϕ_3 is exposed from $c_{5,\pi(j)} = w\phi_3 + s_{\pi(j)}(h_0 + h_1\pi(j))$. From ϕ_3 and $k_{5,i} = r_i$, we can compute $A_i\boldsymbol{v}^\top$ from $k_{4,i}$ for all $i \in [1, m]$. From all these shares $A_i\boldsymbol{v}^\top$ of the LSS scheme, we can compute the shared secret $v_1 = r\phi_2$. From this and $k_3 = r$, we know ϕ_2. From ϕ_2, w and $c_1 = s$, we obtain ϕ_1 from c_3. From s and $s\eta$, we know η. From ϕ_1, r, η, and $k_2 = u$, we can extract α from k_1. □

7.2 Fully-Secure ABE with Short Ciphertexts

Our encoding for KP-ABE with short ciphertexts is shown as Scheme 5. This scheme is bounded ABE where we restrict the maximum size for attribute sets S, denoted by T, while no further restriction is required. We can see that the ciphertext contains only 6 elements. The scheme is a reminiscent of the selectively secure ABE of [2].

Pair Encoding Scheme 5: KP-ABE with Short Ciphertexts

$\mathsf{Param}(T)$ $\rightarrow T+6$. Denote $\boldsymbol{h} = (h_0, h_1, \ldots, h_{T+1}, \phi_1, \phi_2, \phi_3, \eta)$.

$\mathsf{Enc1}(A,\pi) \rightarrow$ For LSS $A \in \mathbb{Z}_N^{m\times k}$, $\pi : [1,m] \rightarrow \mathbb{Z}_N$ (π needed not be injective).
$\boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = (k_1, k_2, k_3, \{k_{4,i}, k_{5,i}, \boldsymbol{k}_{6,i}\}_{i\in[1,m]})$:

$$\left\{\begin{array}{l} k_1 = \alpha + r\phi_1 + u\eta, \qquad k_2 = u, \qquad k_3 = r, \\ k_{4,i} = A_i\boldsymbol{v}^\top + r_i\phi_3, \qquad k_{5,i} = r_i, \\ \boldsymbol{k}_{6,i} = \Big(r_ih_0, r_i(h_2 - h_1\pi(i)), \ldots, r_i(h_{T+1} - h_1\pi(i)^T)\Big) \end{array}\right\}$$

where $v_1 = r\phi_2$, $\boldsymbol{r} = (r, u, r_1, \ldots, r_m, v_2, \ldots, v_k)$, $\boldsymbol{v} = (v_1, \ldots, v_k)$.

$\mathsf{Enc2}(S)$ $\rightarrow$ For $S \subseteq \mathbb{Z}_N$ such that $|S| \le T$,
let a_i be the coefficient of z^i in $p(z) := \prod_{y\in S}(z-y)$.
$\boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = (c_1, c_2, c_3, c_4, c_5, c_6)$:

$$\left\{\begin{array}{lll} c_1 = s, \quad c_2 = s\eta, & & c_3 = s\phi_1 + w\phi_2, \\ c_4 = w, \quad c_5 = w\phi_3 + \tilde{s}(h_0 + h_1a_0 + \cdots + h_{T+1}a_T), & & c_6 = \tilde{s} \end{array}\right\}$$

where $\boldsymbol{s} = (s, w, \tilde{s})$.

Correctness. When $R((A,\pi),S) = 1$, let $I = \{\, i \in [1,m] \mid \pi(i) \in S \,\}$, we have reconstruction coefficients $\{\mu_i\}_{i\in I}$ such that $\sum_{i\in I}\mu_iA_i\boldsymbol{v}^\top = v_1 = r\phi_2$. Hence, we have the following linear combination of the $k_\iota c_j$ terms:

$$k_1c_1 - k_2c_2 - k_3c_3 + \sum_{i\in I}\mu_i(k_{4,i}c_4 - k_{5,i}c_5 + (\boldsymbol{k}_{6,i}(1,\boldsymbol{a})^\top)c_6) = \alpha s - rw\phi_2 + \sum_{i\in I}\mu_i(A_i\boldsymbol{v}^\top w) = \alpha s,$$

where $(1,\boldsymbol{a}) := (1, a_1, \ldots, a_T)$ and a_i is the coefficient of z^i in $p(z) = \prod_{y\in S}(z-y)$, and note that

$$\begin{aligned} \boldsymbol{k}_{6,i}(1,\boldsymbol{a})^\top &= r_i\Big(h_0 + (h_2 - h_1\pi(i))a_1 + \cdots (h_{T+1} - h_1\pi(i)^T)a_T\Big) \\ &= r_i\Big(h_0 + h_2a_1 + \cdots + h_{T+1}a_T - h_1(p(\pi(i)) - a_0)\Big) = r_i(h_0 + h_1a_0 + \cdots + h_{T+1}a_T), \end{aligned} \tag{7}$$

for which we use the fact that $p(y) = 0$ iff $y \in S$, and that $\pi(i) \in S$, hence $p(\pi(i)) = 0$.

We show its doubly selective security in Corollary 15 and 16 below. Now we show that it is not perfectly master-key hiding. The underlying encoding scheme of [2] is also not perfectly master-key hiding by similar argument.

Lemma 10. *Scheme 5 is not perfectly master-key hiding.*

Proof. We consider S and (A,π) such that there exists j where $\pi(j) \in S$ but $R((A,\pi),S) = 0$. Now we will show how to compute α from $\boldsymbol{k}_{(A,\pi)}(\alpha, \boldsymbol{r}, \boldsymbol{h})$ and $\boldsymbol{c}_S(\boldsymbol{s}, \boldsymbol{h})$. Firstly from $\boldsymbol{k}_{6,j}$ and the fact that $\pi(j) \in S$, we can compute $r_j(h_0 + h_1a_0 + \cdots + h_{T+1}a_T)$ using Eq. (7). From this and $k_{5,j} = r_j$, we know $h_0 + h_1a_0 + \cdots + h_{T+1}a_T$. Now since we know $c_6 = \tilde{s}$ and $c_4 = w$, the term ϕ_3 is exposed from c_5. From this point on, the proof proceeds in exactly the same way as the proof of Lemma 9. □

7.3 Key-Policy over Doubly Spatial Encryption Scheme

KP-DSE generalizes KP-ABE in such a way that each "atomic" relation is generalized from equality relation (of which the corresponding FE scheme is IBE) to a so-called doubly spatial relation (of which the corresponding FE scheme is doubly spatial encryption [19]). We begin with definitions for affine spaces.

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

Doubly Spatial Encryption. Doubly-spatial predicate [19] is indexed by the full dimension of spaces, denoted by n. In doubly-spatial encryption [19], we have a key and a ciphertext associated to affine spaces $X \in \mathsf{AffM}(\mathbb{Z}_N^{n\times *})$ and $Y \in \mathsf{AffM}(\mathbb{Z}_N^{n\times *})$ respectively. The relation is defined by $R^{\mathsf{DS}}(X, Y) = 1$ if and only if $\mathsf{AffSp}(X) \cap \mathsf{AffSp}(Y) \neq \emptyset$, *i.e.,* both affine spaces are intersected.

Definition 5 (Key-Policy over Primitives)**.** Let us consider a predicate R defined over domain $\mathbb{X}\times\mathbb{Y}$. Consider a matrix $A \in \mathbb{Z}_N^{m\times k}$ for a linear secret sharing scheme. We associate a key attribute $X^{(i)} \in \mathbb{X}$ to row i of A and call $\mathbb{A} = (A; X^{(1)}, \ldots, X^{(m)})$ an access structure over $\mathbb{X}$. Let $\mathcal{A}_{\mathbb{X}}$ be the universe of considered access structures over $\mathbb{X}$. For a set $\Omega = \{Y^{(1)}, \ldots, Y^{(t)}\} \subseteq \mathbb{Y}$, we define $Q_\Omega = \{i \in [1, m] \mid \exists Y^{(j)} \in \Omega \text{ s.t. } R(X^{(i)}, Y^{(j)}) = 1\}$. We define a new relation $\tilde{R}$ over $\mathcal{A}_{\mathbb{X}} \times 2^{\mathbb{Y}}$ as $\tilde{R}(\mathbb{A}, S) = 1$ if and only if A accepts Q_Ω. We call $\tilde{R}$ the predicate of key-policy over R.

Remark 5. We note that the above definition of key-policy over primitives has an *unbounded* feature in the sense that m, k (for key) and t (for ciphertext) can be arbitrary.

Our Encoding for KP-DSE. Let R^{KPDS} be the predicate of key-policy over R^{DS}. We propose its encoding scheme as Scheme 6. Our scheme is based on a combination of the unbounded KP-ABE scheme of Rouselakis and Waters [29] and doubly spatial encryption of Hamburg [19]. Note that both schemes were proved only selectively secure.

Pair Encoding Scheme 6: KP-DSE

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

$\mathsf{Enc1}(\mathbb{A})$ For $\mathbb{A} = (A; X^{(1)}, \ldots, X^{(m)})$ where $A \in \mathbb{Z}_N^{m\times k}$ and $X^{(i)} \in \mathsf{AffM}(\mathbb{Z}_N^{n\times d_i})$

$\to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = (k_1, k_2, k_3, \{k_{4,i}, k_{5,i}, \boldsymbol{k}_{6,i}\}_{i\in[1,m]})$:

$$\begin{Bmatrix} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_{4,i} = A_i\boldsymbol{v}^\top + r_i\phi_3, & k_{5,i} = r_i, & \boldsymbol{k}_{6,i} = r_i\bar{\boldsymbol{h}}X^{(i)} \end{Bmatrix}$$

where $v_1 := r\phi_2$ and $\boldsymbol{r} = (r, u, r_1, \ldots, r_m, v_2, \ldots, v_k)$, $\boldsymbol{v} := (v_1, v_2, \ldots, v_k)$.

$\mathsf{Enc2}(\Omega)$ For $\Omega = \{Y^{(1)}, \ldots, Y^{(t)}\}$ where $Y^{(j)} \in \mathsf{AffM}(\mathbb{Z}_N^{n\times f_j})$

$\to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = (c_1, c_2, c_3, c_4, \{\boldsymbol{c}_{5,j}, c_{6,j}\}_{j\in[1,t]})$:

$$\begin{Bmatrix} c_1 = s, & c_2 = s\eta, & c_3 = s\phi_1 + w\phi_2, \\ c_4 = w, & \boldsymbol{c}_{5,j} = (w\phi_3, \mathbf{0}) + s_j\bar{\boldsymbol{h}}Y^{(j)}, & c_{6,j} = s_j \end{Bmatrix}$$

where $\boldsymbol{s} = (s, w, s_1, \ldots, s_t)$.

Correctness. The correctness can be shown as follows. If $R^{\mathsf{KPDS}}(\mathbb{A}, Y) = 1$, we can compute a reconstruction coefficients $\{\mu_j\}_{j\in Q_\Omega}$ such that $\sum_{j\in Q_\Omega} \mu_j A_j \boldsymbol{v}^\top = v_1$. For $i \in Q_\Omega$, we pick Y_{j_i} such that $R^{\mathsf{DS}}(X^{(i)}, Y^{(j_i)}) = 1$. For such a pair, the affine spaces intersect, hence there exists an algorithm that computes $\boldsymbol{\gamma}^{(i)}, \boldsymbol{\delta}^{(i)}$ such that $X^{(i)}(\boldsymbol{\gamma}^{(i)})^\top = Y^{(j_i)}(\boldsymbol{\delta}^{(i)})^\top$. Therefore, we have the following bilinear combination:

$$k_1 c_1 - k_2 c_2 - k_3 c_3 + \sum_{j \in Q_\Omega} \mu_j \Big(k_{4,i} c_4 - k_{5,i} \boldsymbol{c}_{5,j} (\boldsymbol{\delta}^{(i)})^\top + \boldsymbol{k}_{6,i} (\boldsymbol{\gamma}^{(i)})^\top c_{6,j} \Big) = \alpha s.$$

Definition 6 ((n,t)-EDHE3 Assumption)**.** The (n,t)-Expanded Diffie-Hellman Exponent Assumption-3 in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, c, b_1, \ldots, b_t, z \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$ (for simplicity). Suppose that an adversary is given a target element $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of

$$\begin{aligned}
& g, g^a, g^c, g^{c/z} \\
\forall_{j \in [1,t]} \quad & g^{b_j} \\
\forall_{i \in [1,n],\ j,j' \in [1,t], j \neq j'} \quad & g^{a^i b_j / b_{j'}^2}, g^{a^n c b_j / b_{j'}} \\
\forall_{i \in [1,2n],\ j \in [1,t]} \quad & g^{a^i c b_j}, \\
\forall_{i \in [1,2n], i \neq n+1,\ j \in [1,t],} \quad & g^{a^i c / b_j}, \\
\forall_{i \in [1,2n],\ j,j' \in [1,t], j \neq j'} \quad & g^{a^i c b_j / b_{j'}^2}, \\
\forall_{i \in [1,n+1],\ j \in [1,t],} \quad & g^{a^i / b_j^2}, \\
\forall_{i \in [n+1,2n],\ j,j' \in [1,t],} \quad & g^{a^i c^2 b_j / b_{j'}}
\end{aligned}$$

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

Theorem 11. *Scheme 6 is selectively master-key hiding under the (n,t)-EDHE3 assumption with tight reduction, where n is the full dimension of spaces in the one-ciphertext query $\Omega^\star$ and t is the number of spaces in $\Omega^\star$.* (The proof is in §C.1).

Definition 7 ((n,m,k)-EDHE4 Assumption)**.** The (n,m,k)-Expanded Diffie-Hellman Exponent Assumption-4 in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, x, c, b_1, \ldots, b_m, \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$ (for simplicity). Suppose that an adversary is given a target element $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of

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

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

Theorem 12. *Scheme 6 is co-selectively master-key hiding under the (n,m,k)-EDHE4 assumption with tight reduction, where n is the full dimension of spaces in the one-key query $\mathbb{A}^\star$ and (m,k) is the matrix dimension of the access matrix A of $\mathbb{A}^\star$.* (The proof is in §C.2).

Intuition for the proofs. Our scheme is naturally designed to contain two layers: key-policy layer and doubly spatial layer. This generalizes the scheme of Rouselakis and Waters [29] KP-ABE which contains key-policy layer and attribute layer. The doubly spatial layer is then implemented using the Hamburg's DSE scheme [19]. For the selectively master-key hiding of our KP-DSE, we use the selective security proof structure of KP-ABE by [29] in the key-policy layer, and the *co-selective* security proof of DSE in the doubly spatial layer. The fact that we use *co-selective* proof of DSE reflects from the fact that we organize the DSE scheme of [19] in a dual manner. We note that there has been no known co-selective security proof for the DSE of [19] (to the best of our knowledge), hence we essentially give a new proof technique for it. Our approach generalizes the selective proof for the attribute layer of *CP-ABE* of [29]. Moreover, in order to cope with the unbounded nature of KP-DSE, the proof will utilize the "individual randomness" techniques employed to achieve unbounded primitives in [36, 25, 29].

On the other hand, for the co-selectively master-key hiding of our KP-DSE, we use the selective security proof of the dual of KP-ABE, which is *CP-ABE* by [29], in the key-policy layer. In the doubly spatial layer, we can use the selective security proof technique for DSE in [19], albeit again generalized to have "individual randomness". Finally, we note that in order to cope with master-key hiding notion, we also must use the trick regarding randomizers described in Remark 4.

7.4 Implications from KP-DSE

Implication to Unbounded KP-ABE. An unbounded KP-ABE (with large universes) can be obtained as a special case of KP-DSE by simply setting the doubly special relation to be the equality relation. More precisely, we define maps F, G where F maps an access structure $\mathbb{A}$ over attributes in universe $\mathbb{Z}_N$ (associated to a key for KP-ABE) to an access structure over affine spaces and G maps a set S of attributes (associated to a ciphertext for KP-ABE) to an affine space. Let $\mathbb{A} = (A; x_1, \dots, x_m)$ where $x_i \in \mathbb{Z}_p$.

$$F : \mathbb{A} = (A, \pi) \mapsto \mathbb{A}' = (A; X^{(1)}, \dots, X^{(m)}), \qquad G : S = \{y_1, \cdots, y_t\} \mapsto \{Y^{(1)}, \dots, Y^{(t)}\}$$

where $X^{(i)} = (1, \pi(i))^\top, Y^{(i)} = (1, y_i)^\top$. It can be verified that $R^{\mathsf{KPDS}}(F(\mathbb{A}), G(S)) = 1$ if and only if $R^{\mathsf{KP\text{-}ABE}}(\mathbb{A}, S) = 1$. We note that in this case the dimension parameter of KP-DSE is $n = 1$.

Applying this embedding to our encoding for KP-DSE (Scheme 6), we obtain exactly our encoding scheme for unbounded KP-ABE with large universes (Scheme 4), described in §7.1. Hence, we obtain the following security theorem for Scheme 4.

Corollary 13. *Scheme 4 is selectively master-key hiding under the* $(1, |S^\star|)$*-EDHE3 assumption with tight reduction, where* $S^\star$ *is the one-ciphertext query.*

Corollary 14. *Scheme 4 is co-selectively master-key hiding under the* $(1, m, k)$*-EDHE4 assumption with tight reduction, where* (m, k) *is the matrix dimension of the access matrix* $A^\star$ *of the one-key query* $(A^\star, \pi^\star)$*.*

Implication to KP-ABE with Short Ciphertexts. Recall first that in KP-ABE with short ciphertexts, we require the bound T on the maximum size of attribute set S. This primitive can be obtained as a special case of KP-DSE by setting $n = T + 1$ and $|\Omega| = t = 1$. That is, we will use only one element Y of the ciphertext and set the doubly spatial relation to be the set inclusion relation (*à la* ID-based broadcast encryption). More precisely, we define maps

$$F : \mathbb{A} = (A, \pi) \mapsto \mathbb{A}' = (A; X^{(1)}, \dots, X^{(m)}), \qquad G : S \mapsto (1, a_0, \dots, a_T)^\top$$

where

$$X^{(i)} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -\pi(i) & -\pi(i)^2 & \cdots & -\pi(i)^T \\ 0 & 1 & & & \\ 0 & & 1 & & \\ \vdots & & & \ddots & \\ 0 & & & & 1 \end{pmatrix} \in \mathsf{AffM}(\mathbb{Z}_N^{(T+1)\times T})$$

and a_i is coefficient in $p(z) = \prod_{y\in S}(z-y) = a_0 + a_1 z + \cdots + a_T z^T$. It can be verified that $R^{\mathsf{KPDS}}(F(\mathbb{A}), G(S)) = 1$ if and only if $R^{\mathsf{KP\text{-}ABE}}(\mathbb{A}, S) = 1$.

Applying this embedding to our encoding for KP-DSE (Scheme 6), we obtain exactly our encoding scheme for KP-ABE with short ciphertexts (Scheme 5), described in §7.2. Hence, we obtain the following security theorem for Scheme 5.

Corollary 15. *Scheme 5 is selectively master-key hiding under the $(T+1, 1)$-EDHE3 assumption with tight reduction, where T is the maximum size of attribute set associated to ciphertext.*

Corollary 16. *Scheme 5 is co-selectively master-key hiding under the $(T+1, m, k)$-EDHE4 assumption with tight reduction, where T is the maximum size of attribute set associated to ciphertext and (m, k) is the matrix dimension of the access matrix $A^\star$ of the one-key query $(A^\star, \pi^\star)$.*

Implication to Doubly Spatial Encryption. We obtain this by just neglecting the key-policy portion of the scheme. This gives a fully-secure doubly spatial encryption with $O(q_1)$ reduction.

8 Dual Scheme Conversion

8.1 Generic Dual Scheme Conversion for Perfectly Secure Encoding

In this section, we will describe a simple tool for converting a perfectly secure pair encoding of any predicate R to another perfectly secure pair encoding of the *dual* predicate $\bar{R}$. A most well-known example is that CP-ABE is the dual primitive of KP-ABE. The definition is naturally defined as follows. For a predicate $R : \mathbb{X} \times \mathbb{Y}$ its dual predicate is defined by $\bar{R} : \bar{\mathbb{X}} \times \bar{\mathbb{Y}}$ where $\bar{\mathbb{X}} = \mathbb{Y}, \bar{\mathbb{Y}} = \mathbb{X}$ and $\bar{R}(X, Y) := R(Y, X)$. Applications of the conversion will be shown when we revisit Lewko *et al.* ABE in §9.1.

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

- $\overline{\mathsf{Enc1}}(X, N)$: Run $(\boldsymbol{c}'(\boldsymbol{s}', \boldsymbol{h}); w_2) \leftarrow \mathsf{Enc2}(X, N)$, parse $\boldsymbol{s}' = (s', \ldots)$, set $\boldsymbol{k} := (\boldsymbol{c}', \alpha + \phi s')$, $\boldsymbol{r} := \boldsymbol{s}'$, and output $(\boldsymbol{k}(\alpha, \boldsymbol{r}, \bar{\boldsymbol{h}}); w_2)$, where we treat α as a new variable.

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

The correctness can be verified as follows. If $\bar{R}(X, Y) = 1$, then $R(Y, X) = 1$, hence from the correctness of P_R, we can compute $\alpha' s' = (\phi s) s'$. From that we obtain $(\alpha + \phi s')(s) - (\phi s) s' = \alpha s$.

Lemma 17. *If P_R is a perfectly master-key hiding encoding scheme for R, then $\mathsf{P}_{\bar{R}}$ is a perfectly master-key hiding encoding scheme for $\bar{R}$.*

Proof. If $\bar{R}(X,Y)=0$, then $R(Y,X)=0$. Hence, from perfect security we have that $\alpha'=\phi s$ is hidden. Dividing the known s term, we have ϕ is hidden, which means $\phi s'$ is hidden, and this masks α to be hidden in $\alpha+\phi s'$. □

8.2 Fully Secure Dual FE for Regular Languages

We do not know a generic conversion in the case of doubly selective master-key hiding encoding. To this end, we will directly construct and prove security for dual FE for given primitives. We demonstrate here for our focus predicate, namely, we construct a fully secure scheme for the dual FE primitive for regular languages, where ciphertexts are associated with a DFA machine M, and keys are associated with a string of any sizes. This encoding scheme is shown as Scheme 7.

Pair Encoding Scheme 7: Dual FE for regular languages

Param $\to$ 9. Denote $\boldsymbol{h}=(h_0,h_1,h_2,h_3,h_4,\phi_1,\phi_2,\eta,\phi)$.

Enc1(w) $\to$ For $w\in(\mathbb{Z}_N)^*$, let $\ell=|w|$, and parse $w=(w_1,\dots,w_\ell)$.

$$\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})=(k_1,k_2,k_3,k_4,\{k_{5,i}\}_{i\in[0,\ell]},\{k_{6,i}\}_{i\in[1,\ell]},k_7):$$

$$\left\{\begin{array}{lll} k_1=r, & k_2=\alpha+r\phi+u\eta, & k_3=-r\phi_1+r_\ell\phi_2,\\ k_4=r_0h_0, & k_{5,i}=r_i, & k_{6,i}=r_{i-1}(h_1+h_2w_i)+r_i(h_3+h_4w_i),\\ k_7=u & & \end{array}\right\}$$

where $\boldsymbol{r}=(r,r_0,r_1,\dots,r_\ell,u)$.

Enc2(M) $\to$ For any DFA $M=(Q,\mathbb{Z}_N,\mathcal{T},q_0,q_{n-1})$, where $n=|Q|$, let $m=|\mathcal{T}|$, and parse $\mathcal{T}=\{(q_{x_t},q_{y_t},\sigma_t)|t\in[1,m]\}$.

$$\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})=(c_1,c_2,c_3,c_4,c_5,\{c_{6,t},c_{7,t},c_{8,t}\}_{t\in[1,m]}):$$

$$\left\{\begin{array}{lll} c_1=s\phi+v\phi_1, & c_2=s, & c_3=v,\\ c_4=s_0, & c_5=-u_0+s_0h_0, & c_{6,t}=s_t,\\ c_{7,t}=u_{x_t}+s_t(h_1+h_2\sigma_t), & c_{8,t}=-u_{y_t}+s_t(h_3+h_4\sigma_t) & c_9=s\eta \end{array}\right\}$$

where $u_{n-1}:=\phi_2v$ and $\boldsymbol{s}=(v,s,s_0,s_1,\dots,s_m,\{u_x\}_{q_x\in Q\smallsetminus\{q_{n-1}\}})$.

The correctness can be verified similarly (and dually) to Scheme 3. We omitted it here. The scheme resembles the generic dual conversion but with some tweaks regarding the randomizer technique of Remark 4. The intuition for the security proof is to use the proof of selective master-key hiding of encoding Scheme 3 for co-selective master-key hiding of encoding Scheme 7, and vice versa. We will use slightly modified security assumptions which we describe in §D.1 and §D.2.

Lemma 18. *Scheme 7 is not perfectly master-key hiding.*

Proof. This is similar to the disproof of perfectly master-key hiding security for Scheme 3 (*cf.* Lemma 6). □

Theorem 19. *Scheme 7 is selectively master-key hiding under the (n,m)-EDHE2-Dual assumption with tight reduction, where $n=|Q|$ is the number of states, $m=|\mathcal{T}|$ is the number of transitions of the one-ciphertext query DFA $M^\star$.* (The proof is in §D.1).

Theorem 20. *Scheme 7 is co-selectively master-key hiding under the ℓ-EDHE1-Dual assumption with tight reduction, where ℓ is the length of the one-key query $w^\star$.* (The proof is in §D.2).

9 Unifying and Improving Existing Schemes

In this section, we revisit some important existing primitives and their implementation schemes, and cast them into our framework by extracting their underlying pair encoding schemes. All the revisited schemes are based on the classical dual system approach, hence we re-prove the security by simply showing the perfect master-key hiding security of their encoding schemes. We then give some improvements over these schemes.

9.1 Attribute-Based Encryption

Small-Universe KP-ABE (Lewko *et al.*). The pair encoding scheme extracted from Lewko *et al.* KP-ABE [27] (or called the LOSTW scheme) is shown as Scheme 8.

Pair Encoding Scheme 8: LOSTW KP-ABE

$\mathsf{Param}(|\mathfrak{U}|) \to |\mathfrak{U}|$. Denote $\boldsymbol{h} = (h_a)_{a\in\mathfrak{U}}$.

$\mathsf{Enc1}(A,\pi) \to$ For LSS $A \in \mathbb{Z}_N^{m\times k}$, and $\pi : [1,m] \to \mathfrak{U}$. ($\pi$ is injective).
$(\{k_{1,i}, k_{2,i}\}_{i\in[1,m]})$:

$$\left\{ \begin{array}{l} k_{1,i} = A_i\boldsymbol{\alpha}^\top + r_i h_{\pi(i)}, \\ k_{2,i} = r_i, \end{array} \right\}$$

where $\boldsymbol{\alpha} := (\alpha, v_2, \dots, v_k)$ and $\boldsymbol{r} = (r_1, \dots, r_m, v_2, \dots, v_k)$.

$\mathsf{Enc2}(Y) \to$ For $Y \subseteq \mathfrak{U}$
$(c_1, \{c_{2,y}\}_{y\in Y})$:

$$\left\{ \begin{array}{c} c_1 = s, \\ c_{2,y} = sh_y \end{array} \right\}$$

where $\boldsymbol{s} = s$.

Pair Encoding Scheme 9: Improved KP-ABE

$\mathsf{Param}(|\mathfrak{U}|) \to |\mathfrak{U}|$. Denote $\boldsymbol{h} = (h_a)_{a\in\mathfrak{U}}$.

$\mathsf{Enc1}(A,\pi) \to$ For LSS $A \in \mathbb{Z}_N^{m\times k}$, and $\pi : [1,m] \to \mathfrak{U}$. ($\pi$ is injective).
$(\{k_{1,i}\}_{i\in[1,m]}, k_2)$:

$$\left\{ \begin{array}{l} k_{1,i} = A_i\boldsymbol{\alpha}^\top + rh_{\pi(i)}, \\ k_2 = r, \end{array} \right\}$$

where $\boldsymbol{\alpha} := (\alpha, v_2, \dots, v_k)$ and $\boldsymbol{r} = (r, v_2, \dots, v_k)$.

$\mathsf{Enc2}(Y) \to$ for $Y \subseteq \mathfrak{U}$
$(c_1, \{c_{2,y}\}_{y\in Y})$:

$$\left\{ \begin{array}{c} c_1 = s, \\ c_{2,y} = sh_y \end{array} \right\}$$

where $\boldsymbol{s} = s$.

The correctness can be shown as follows. When $R((A,\pi),Y) = 1$, we have that there is a set $I \in [1,m]$ such that $\{\, \pi(i) \mid i \in I \,\} \subseteq Y$ and a reconstruction coefficients $\{\mu_i\}_{i\in I}$ such that $\sum_{i\in I} \mu_i A_i \boldsymbol{\alpha}^\top = \alpha$. Therefore, we have the following linear combination of the $k_\iota c_\kappa$ terms:

$$\sum_{i\in I} w_i(k_{1,i}c_1 - k_{2,i}c_{2,\pi(i)}) = \sum_{i\in I} w_i A_i \boldsymbol{\alpha}^\top s = \alpha s. \tag{8}$$

Lemma 21. *Scheme 8 is perfectly master-key hiding.*

Proof. When $R(X,Y) = 0$, we have that (A,π) does not accept Y. For $j = 1, \dots, m$, we consider two cases. If $\pi(j) \notin Y$, then $h_{\pi(j)}$ does not appear anywhere and hence the information on α will not be leaked from k_j. Now consider when $\pi(j) \in Y$. In this case, both r_j and $h_{\pi(j)}$ is available, hence $A_j\boldsymbol{\alpha}^\top$ is known. Now from the lemma of LSSS (similar to Proposition 40), there exists $\boldsymbol{\omega} \in \mathbb{Z}_N^k$ with $\omega_1 \neq 0$ such that $\boldsymbol{\omega}$ is orthogonal to all A_j where $\pi(j) \in Y$. Hence, $A_j\boldsymbol{\alpha}^\top = A_j(\boldsymbol{\alpha}^\top + z\boldsymbol{\omega}^\top)$ for any unknown random $z \in \mathbb{Z}_N$. Therefore, $A_j\boldsymbol{\alpha}^\top$ does not leak the information on α due to $\omega_1 \neq 0$. □

Improved Small-Universe KP-ABE (New). We observe that in our proof for the LOSTW KP-ABE above, we did not use any advantage of r_i being different for all $i = 1, \dots, m$. Hence indeed we can set $r := r_1 = \cdots = r_m$. This makes the key size much shorter (about half-size of the original scheme). Our encoding is written as Scheme 9 and is also perfectly master-key hiding.

Small-Universe CP-ABE (Lewko *et al.*). The encoding scheme extracted from CP-ABE with small universe of Lewko *et al.* [27] is shown as Scheme 10 below. Scheme 10 turns out to be exactly the encoding scheme obtained from the generic dual conversion (of §8) applied to the KP-ABE Scheme 8 shown above. Therefore, we obtain the following.

Lemma 22. *Scheme 10 is perfectly master-key hiding.*

Proof. This is direct from Lemma 17 and Lemma 21. □

Pair Encoding Scheme 10: LOSTW CP-ABE

$\mathsf{Param}(|\mathcal{U}|) \to |\mathcal{U}|+1.$
Denote $\boldsymbol{h} = (\phi, (h_a)_{a\in\mathcal{U}})$.

$\mathsf{Enc1}(X) \to$ For $X \subseteq \mathcal{U}$
$(k_1, \{k_{2,x}\}_{x\in X}, k_3)$:

$$\left\{\begin{aligned} k_1 &= r, \\ k_{2,x} &= r h_x \\ k_3 &= \alpha + \phi r \end{aligned}\right\}$$

where $\boldsymbol{r} = r$.

$\mathsf{Enc2}(A,\pi) \to$ For LSS $A \in \mathbb{Z}_N^{m\times k}$, and
$\pi : [1,m] \to \mathcal{U}$. ($\pi$ is injective).
$(\{c_{1,i}, c_{2,i}\}_{i\in[1,m]}, c_3)$:

$$\left\{\begin{aligned} c_{1,i} &= \phi A_i \boldsymbol{v}^\top + s_i h_{\pi(i)}, \\ c_{2,i} &= s_i, \\ c_3 &= s \end{aligned}\right\}$$

where $\boldsymbol{v} := (s, v_2, \ldots, v_k)$ and
$\boldsymbol{s} = (s, s_1, \ldots, s_m, v_2, \ldots, v_k)$.

Pair Encoding Scheme 11: Improved CP-ABE

$\mathsf{Param}(|\mathcal{U}|) \to |\mathcal{U}|+1.$
Denote $\boldsymbol{h} = (\phi, (h_a)_{a\in\mathcal{U}})$.

$\mathsf{Enc1}(X) \to$ For $X \subseteq \mathcal{U}$
$(k_1, \{k_{2,x}\}_{x\in X}, k_3)$:

$$\left\{\begin{aligned} k_1 &= r, \\ k_{2,x} &= r h_x \\ k_3 &= \alpha + \phi r \end{aligned}\right\}$$

where $\boldsymbol{r} = r$.

$\mathsf{Enc2}(A,\pi) \to$ For LSS $A \in \mathbb{Z}_N^{m\times k}$, and
$\pi : [1,m] \to \mathcal{U}$. ($\pi$ is injective).
$(\{c_{1,i}\}_{i\in[1,m]}, c_2, c_3)$:

$$\left\{\begin{aligned} c_{1,i} &= \phi A_i \boldsymbol{v}^\top + \tilde{s} h_{\pi(i)}, \\ c_2 &= \tilde{s}, \\ c_3 &= s \end{aligned}\right\}$$

where $\boldsymbol{v} := (s, v_2, \ldots, v_k)$ and
$\boldsymbol{s} = (s, \tilde{s}, v_2, \ldots, v_k)$.

Improved Small-Universe CP-ABE (New). Similarly, we obtain a new improved encoding scheme for CP-ABE shown as Scheme 11. The ciphertext size is about half of the LOSTW CP-ABE scheme.

Large-Universe KP-ABE (New). In ABE with large universe, the attribute universe $\mathcal{U}$ is of exponential size (in the security parameter). We presented a fully secure unbounded scheme in Section 7.4 under new static assumptions. In this section, we present a new scheme for free, but the semantic requires bounded sizes $m_{\max}$ of the policy sizes (the number of rows in the LSS matrix A) and $t_{\max}$ of the attribute set Y. The construction is based on cover-free families [11, 21]. Their definition and existence are given as follows.

Definition 8 (Cover-Free Families [11])**.** Consider a set system $(B, \{S_i\}_{i\in\mathcal{U}})$, where $S_i \subset B$. It is a f-cover-free family if for all $S_{i_1}, \ldots, S_{i_{f+1}}$ we have $S_{i_{f+1}} \not\subseteq \bigcup_{j=1}^{f} S_{i_j}$.

Proposition 23 ([21])**.** *There exists a deterministic polynomial-time algorithm that, on input of integers f, n, returns a set system $(B, \{S_i\}_{i\in\mathcal{U}})$ that is f-cover-free family where $|B| \leq 16f^2 \log n$, $|\mathcal{U}| = n$, and for all $i \in \mathcal{U}$, we have $|S_i| = |B|/4f$.*

We are now ready to describe the encoding scheme for KP-ABE with large universe as Scheme 12. For attribute universe of size n, which is exponential, and bounds $m_{\max}, t_{\max}$, we use $(m_{\max} + t_{\max} -$

1)-cover-free family $(B, \{S_i\}_{i\in\mathcal{U}})$ with $|\mathcal{U}| = n$, of which the existence is stated in Proposition 23. The idea is that even $|\mathcal{U}|$ is of exponential size, $|B|$ is still of polynomial size, due to this proposition. The correctness can be shown exactly as in the small universe scheme (Eq.(8)).

Pair Encoding Scheme 12: New KP-ABE with large universe	
$\mathsf{Param}(n, m_{\max}, t_{\max}) \to$	$|B|$, where $(B, \{S_i\}_{i\in\mathcal{U}})$ is a $(m_{\max} + t_{\max} - 1)$-cover-free family with $|\mathcal{U}| = n$. Denote $\boldsymbol{h} = (h_i)_{i\in B}$.
$\mathsf{Enc1}(A, \pi)$ $\to$	For LSS $A \in \mathbb{Z}_N^{m\times k}$, $\pi : [1, m] \to \mathcal{U}$, where $m \leq m_{\max}$ $(\{k_{1,i}\}_{i\in[1,m]}, k_2)$: $\left\{ k_{1,i} = A_i\boldsymbol{\alpha}^\top + r \sum_{j\in S_{\pi(i)}} h_j, \quad k_2 = r \right\}$ where $\boldsymbol{\alpha} := (\alpha, v_2, \dots, v_k)$ and $\boldsymbol{r} = (r, v_2, \dots, v_k)$.
$\mathsf{Enc2}(Y)$ $\to$	For $Y \subseteq \mathcal{U}$ where $|Y| \leq t_{\max}$ $(c_1, \{c_{2,y}\}_{y\in Y})$: $\left\{ c_1 = s, \quad c_{2,y} = s \sum_{j\in S_y} h_j \right\}$ where $\boldsymbol{s} = s$.

Theorem 24. *Scheme 12 is perfectly master-key hiding.*

Proof. When $R(X, Y) = 0$, we have that (A, π) does not accept Y. For $j = 1, \dots, m$, we consider two cases. We first consider the case $\pi(j) \notin Y$. In this case, we have

$$S_{\pi(j)} \not\subseteq \Big(\bigcup_{\iota\in[1,m]\setminus\{j\}} S_{\pi(\iota)} \Big) \cup \Big(\bigcup_{i\in Y} S_i \Big),$$

due to the property of the cover-free family, where we observe that the number of sets in the right-hand side is $m + t - 1 \leq m_{\max} + t_{\max} - 1$ and that $S_{\pi(j)}$ is not one of these since π is injective and $\pi(j) \notin Y$. Therefore, there will be at least one h_ℓ which does not appear elsewhere except in k_j and this will hide the information on α from being leaked from k_j. Next, we consider the case $\pi(j) \in Y$. In this case, $r, s, s(\sum_{i\in S_{\pi(j)}} h_i)$ is available, hence $A_j\boldsymbol{\alpha}^\top$ is trivially known. Now from the lemma of LSSS, there exists $\boldsymbol{\omega} \in \mathbb{Z}_N^k$ with $\omega_1 \neq 0$ such that $\boldsymbol{\omega}$ is orthogonal to all A_j where $\pi(j) \in Y$. Hence, $A_j\boldsymbol{\alpha}^\top = A_j(\boldsymbol{\alpha}^\top + z\boldsymbol{\omega}^\top)$ for any random z. Therefore, $A_j\boldsymbol{\alpha}^\top$ does not leak the information on α due to $\omega_1 \neq 0$. □

Large-Universe CP-ABE (New). By applying the generic dual scheme conversion to Scheme 12, we obtain a perfectly master-key hiding pair encoding scheme for CP-ABE with large universes as Scheme 13.

Pair Encoding Scheme 13: New CP-ABE with large universe	
$\mathsf{Param}(n, m_{\max}, t_{\max})$	$\to |B|$, where $(B, \{S_i\}_{i\in\mathcal{U}})$ is a $(m_{\max} + t_{\max} - 1)$-cover-free family with $|\mathcal{U}| = n$. Denote $\boldsymbol{h} = (h_i)_{i\in B}$.
$\mathsf{Enc1}(X)$	For $X \subseteq \mathcal{U}$ where $|X| \le t_{\max}$ $\to (k_1, \{k_{2,x}\}_{x\in X}, k_3)$: $\left\{ k_1 = r, \quad k_{2,x} = r \sum_{j\in S_x} h_j, \quad k_3 = \alpha + \phi r \right\}$ where $\boldsymbol{r} = r$.
$\mathsf{Enc2}(A, \pi)$	For LSS $A \in \mathbb{Z}_N^{m\times k}$, $\pi : [1, m] \to \mathcal{U}$, where $m \le m_{\max}$ $\to (\{c_{1,i}\}_{i\in[1,m]}, c_2, c_3)$: $\left\{ c_{1,i} = \phi A_i \boldsymbol{v}^\top + \tilde{s} \sum_{j\in S_{\pi(i)}} h_j, \quad c_2 = \tilde{s}, \quad c_3 = s \right\}$ where $\boldsymbol{v} := (s, v_2, \dots, v_k)$ and $\boldsymbol{s} = (s, \tilde{s}, v_2, \dots, v_k)$.

9.2 Doubly Spatial Encryption and Negated Spatial Encryption

Doubly Spatial Encryption. In Section 7, we constructed key-policy over doubly spatial encryption and mentioned the mere doubly spatial encryption as a special case. In this section, we point out indeed that the encoding scheme extracted from the doubly spatial encryption scheme of Hamburg [19] can be proved perfectly master-key hiding, hence we can obtain fully secure doubly spatial encryption for free (but now with normal reduction). The scheme is shown as Scheme 14.

Pair Encoding Scheme 14: Doubly Spatial Encryption	
$\mathsf{Param}(n) \to n + 1$. Denote $\boldsymbol{h} = (h_0, h_1, \dots, h_n)$.	
$\mathsf{Enc1}(X)$	For $X \in \mathsf{AffM}(\mathbb{Z}_N^{n\times d})$ $\to \Big(\boldsymbol{k}_1 = (\alpha, \mathbf{0}) + r\boldsymbol{h}X, \quad k_2 = r \Big)$, where $\boldsymbol{r} = r$.
$\mathsf{Enc2}(Y)$	For $Y \in \mathsf{AffM}(\mathbb{Z}_N^{n\times f})$ $\to \Big(c_1 = s, \quad \boldsymbol{c}_2 = s\boldsymbol{h}Y, \Big)$, where $\boldsymbol{s} = s$.

The correctness can be shown as follows. When $R(X, Y) = 1$ (*i.e.*, $\mathsf{AffSp}(X) \cap \mathsf{AffSp}(Y) \neq \emptyset$), we can compute affine vectors $\boldsymbol{w} \in \mathsf{Aff}(\mathbb{Z}_N^d), \boldsymbol{z} \in \mathsf{Aff}(\mathbb{Z}_N^f)$ such that $X\boldsymbol{w}^\top = Y\boldsymbol{z}^\top$. Therefore, we have

$$\boldsymbol{k}_1\boldsymbol{w}^\top c_1 - k_2\boldsymbol{c}_2\boldsymbol{z}^\top = ((\alpha, \mathbf{0}) + r\boldsymbol{h}X)\boldsymbol{w}^\top s - r(s\boldsymbol{h}Y)\boldsymbol{z}^\top = \alpha s.$$

We prove its perfectly master-key hiding security as follows.

Theorem 25. *Scheme 14 is perfectly master-key hiding.*

Proof. The encoding construction implies a system of equations with unknown $\alpha, \boldsymbol{h}$:

$$\left(\begin{array}{c|c} \begin{matrix} 1 \\ \mathbf{0}_d^\top \end{matrix} & rX^\top \\ \hline \mathbf{0}_{f+1}^\top & sY^\top \end{array}\right) \begin{pmatrix} \alpha \\ \boldsymbol{h}^\top \end{pmatrix} = \begin{pmatrix} \boldsymbol{k}_1^\top \\ c_2 \end{pmatrix},$$

where $r (= k_2)$ and $s (= c_1)$ are known. Assume that $R(X, Y) = 0$. We have that there is no affine vectors $\boldsymbol{w} \in \mathsf{Aff}(\mathbb{Z}_N^d), \boldsymbol{z} \in \mathsf{Aff}(\mathbb{Z}_N^f)$ such that $\boldsymbol{w}X^\top = \boldsymbol{z}Y^\top$. This implies that $(1, \mathbf{0}_{n+1})$ is not in the row space of the matrix on the left. Hence α is completely hidden. □

Negated Spatial Encryption. Spatial encryption [6] is a special case of doubly spatial encryption where ciphertext attribute are confined only to vectors, instead of general affine spaces. Negated spatial relation is its negated version. In negated spatial encryption [1], we have a key associated to a vector space $X = V(M, \mathbf{0}) = \{\boldsymbol{w}M^\top \mid \boldsymbol{w} \in \mathbb{Z}_N^{n-1}\}$ where M has rank $n-1$, while a ciphertext is associated to a vector $Y = \boldsymbol{y}$, and $R(X,Y) = 1$ iff $\boldsymbol{y} \notin V(M, \mathbf{0})$. Note that X is a vector space since it passes through the origin. We also note that non-zero inner-product relation is a special case of this relation (*cf.* [1]).

We extract the pair encoding scheme from the selectively secure scheme of [1] into Scheme 15.

Pair Encoding Scheme 15: Negated Spatial Encryption

$\mathsf{Param}(n) \to n$. Denote $\boldsymbol{h} = (h_1, \ldots, h_n)$.

For $M \in \mathbb{Z}_N^{n \times d}$ representing $(n-1)$-rank space $V(M, \mathbf{0})$

$\mathsf{Enc1}(M) \to \Big(k_1 = \alpha + rh_1 \quad \boldsymbol{k}_2 = r\boldsymbol{h}M, \quad k_3 = r\Big)$, where $\boldsymbol{r} = r$.

For $\boldsymbol{y} \in \mathbb{Z}_N^n$

$\mathsf{Enc2}(\boldsymbol{y}) \to \Big(c_1 = s, \quad c_2 = s\boldsymbol{h}\boldsymbol{y}^\top\Big)$, where $\boldsymbol{s} = s$.

To prove correctness and security, we first note that our construction implies a system of equations as follows. Here $M_1^\top$ is the first column of $M^\top$ and $M_{2\to n}^\top$ is the matrix formed by column 2 to n of $M^\top$. Also, $\boldsymbol{y}_{2\to n}$ is similarly defined.

$$\begin{pmatrix} 1 & r & 0 \\ \mathbf{0}^\top & rM_1^\top & rM_{2\to n}^\top \\ 0 & sy_1 & s\boldsymbol{y}_{2\to n} \end{pmatrix} \begin{pmatrix} \alpha \\ \boldsymbol{h}^\top \end{pmatrix} = \begin{pmatrix} \boldsymbol{k}^\top \\ c_2 \end{pmatrix}.$$

The correctness can be verified as follows. Assume $R(X,Y) = 1$ (*i.e.*, $\boldsymbol{y} \notin X$). Since $M_{2\to n}^\top$ has rank $n-1$, we have $s\boldsymbol{y}_{2\to n} \in \mathsf{RowSp}(rM_{2\to n}^\top)$. Hence we can compute $\boldsymbol{w} \in \mathbb{Z}_N^{n-1}$ such that $\boldsymbol{w}M_{2\to n}^\top - \boldsymbol{y}_{2\to n} = \mathbf{0}_{n-1}$. But since $\boldsymbol{y} \notin X$, we have $D := \boldsymbol{w}M_1^\top - y_1 \neq 0$ (otherwise $\boldsymbol{y} \in \mathsf{RowSp}(M^\top)$). Therefore, we can compute

$$k_1c_1 - \boldsymbol{k}_2c_1\boldsymbol{w}^\top D^{-1} + k_3c_2D^{-1} = \alpha s.$$

Theorem 26. *Scheme 15 is perfectly master-key hiding.*

Proof. When $R(X,Y) = 0$, we have that $\boldsymbol{y}$ is in the space $X = V(M, \mathbf{0})$. $\mathbf{0}_{n-1}$ is not in the row space of the matrix that is formed by the last $n-1$ columns of the above matrix. Hence $(1, \mathbf{0}_n)$ is not in the row space of the whole matrix. Therefore α is hidden. □

Acknowledgement. I would like to thank Michel Abdalla, Takahiro Matsuda, Shota Yamada, and anonymous reviewers for their helpful comments on previous versions of this paper. I would like to also thank Hoeteck Wee for fruitful discussions on relations to his work.

A Security Proof of Our Framework

A.1 Proof for Theorem with Tighter Reduction

In this section, we describe the security proof of Theorem 1.

We prove the distinguishability between the consecutive games as in the following lemmata, where we define $\mathsf{G}_j\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)$ to be the advantage of $\mathcal{A}$ in the game G_j. We note that in the final game, the advantage of $\mathcal{A}$ will be 0. Therefore, from these lemmata, we can compute the advantage of $\mathcal{A}$ to be the one shown in the theorem statement. This would conclude the proof.

It remains to prove all the lemmata.

Lemma 27. *If an adversary $\mathcal{A}$ can distinguish $\mathsf{G}_{\mathrm{real}}$ from $\mathsf{G}_{\mathrm{res}}$, we can build an algorithm $\mathcal{B}$ that breaks Assumption 1 or Assumption 2. Indeed, $|\mathsf{G}_{\mathrm{real}}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{\mathrm{res}}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD1}}(\lambda) + \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD2}}(\lambda)$.*

Proof. Assume that there is an adversary $\mathcal{A}$ that produces $X, Y^\star \in \mathbb{Z}_N$ such that $R_{p_2}(X, Y^\star) = 1$ but $R_N(X, Y^\star) = 0$. From the soundness of domain-transferability of R, we can use the corresponding algorithm $\mathcal{F}$ to obtain a factor F of N, such that $p_2|F$. Let $B = N/F$. We can consider two cases: $p_1|B$ or $F = p_1p_2$ and $B = p_3$. The first case, we can break Assumption 1, while in the second case we can break assumption 2, in exactly the same way as described in [23]. □

Lemma 28. *If an adversary $\mathcal{A}$ can distinguish $\mathsf{G}_{\mathrm{res}}$ from G_0, we can build an algorithm $\mathcal{B}$ that breaks Assumption 1. Indeed, $|\mathsf{G}_{\mathrm{res}}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_0\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD1}}(\lambda)$.*

Proof. Our algorithm $\mathcal{B}$ takes in a problem instance (g_1, Z_3, T) for Assumption 2. Its task will be to decide whether $T \in \mathbb{G}_{p_1p_2}$ or $T \in \mathbb{G}_{p_1}$.

Setup. Algorithm $\mathcal{B}$ picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha, \xleftarrow{\$} \mathbb{Z}_N$, and prepares the public key $\mathsf{PK} = (g_1, e(g_1, g_1)^\alpha, Z_3, g_1^{\boldsymbol{h}})$ as usual. It sends PK to the adversary $\mathcal{A}$. Let g_2 be an unknown random generator of $\mathbb{G}_{p_2}$. $\mathcal{B}$ implicitly sets the semi-functional parameter $\hat{\boldsymbol{h}}$ by implicitly defining $\hat{\boldsymbol{h}} \bmod p_2 = \boldsymbol{h} \bmod p_2$. This is properly distributed $\hat{\boldsymbol{h}}$; in particular, it is independent from $\boldsymbol{h} \bmod p_1$ due to the Chinese Remainder Theorem.

Phase 1. When $\mathcal{A}$ makes the j-th private key query, $\mathcal{B}$ generates a normal key as in the construction. This can be done since it knows the master key $\mathsf{msk} = \alpha$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$. $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0,1\}$, runs $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $\boldsymbol{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge $(\boldsymbol{C}, C_0)$ as

$$\boldsymbol{C} = T^{\boldsymbol{c}(\boldsymbol{s}', \boldsymbol{h})}, \qquad C_0 = (e(T, g_1)^\alpha)^{s'} M_b.$$

We claim that it is a properly distributed normal or semi-functional ciphertext. To see this, first we observe that its $\mathbb{G}_{p_1}$ component is

$$g_1^{t_1\boldsymbol{c}(\boldsymbol{s}', \boldsymbol{h})} = g_1^{\boldsymbol{c}(t_1\boldsymbol{s}', \boldsymbol{h})},$$

where we write $T = g_1^{t_1} g_2^{t_2}$. This is properly distributed as in Equation (1) and (3) with $\boldsymbol{s} = t_1\boldsymbol{s}'$ (mod p_1). If $T \in \mathbb{G}_{p_1}$, then it has no $\mathbb{G}_{p_2}$ component hence is a normal ciphertext. If $T \in \mathbb{G}_{p_1p_2}$, then this is a semi-functional ciphertext of which the $\mathbb{G}_{p_2}$ component is

$$g_2^{t_2\boldsymbol{c}(\boldsymbol{s}', \boldsymbol{h})} = g_2^{\boldsymbol{c}(t_2\boldsymbol{s}', \boldsymbol{h})} = g_2^{\boldsymbol{c}(t_2\boldsymbol{s}', \hat{\boldsymbol{h}})},$$

which is properly distributed as in Equation (3) with $\hat{\boldsymbol{s}} = t_2\boldsymbol{s}' \pmod{p_2}$. These values are uniformly distributed since they are not correlated with the terms in modulo p_1 due to the Chinese Remainder Theorem.

Phase 2. $\mathcal{B}$ does the same as in Phase 1.

Guess. The algorithm $\mathcal{B}$ has properly simulated G_{res} if $T \in \mathbb{G}_{p_1}$ and G_0 if $T \in \mathbb{G}_{p_1p_2}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break the assumption 1. $\square$

Lemma 29. *For $k \in [1, q_1]$, if an adversary $\mathcal{A}$ can distinguish $\mathsf{G}_{k-1,3}$ from $\mathsf{G}_{k,1}$, we can build an algorithm $\mathcal{B}$ that breaks Assumption 2. Indeed, $|\mathsf{G}_{k-1,3}\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) - \mathsf{G}_{k,1}\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}}(\lambda)$.*

Proof. Our algorithm $\mathcal{B}$ takes in a problem instance $(g_1, Z_1Z_2, Z_3, W_2W_3, T)$ for Assumption 2. Its task will be to decide whether $T \in \mathbb{G}_{p_1p_2p_3}$ or $T \in \mathbb{G}_{p_1p_3}$.

Setup. Algorithm $\mathcal{B}$ picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha, \xleftarrow{\$} \mathbb{Z}_N$, and prepares the public key $\mathsf{PK} = (g_1, e(g_1, g_1)^{\alpha}, Z_3, g_1^{\boldsymbol{h}})$ as usual. It sends PK to the adversary $\mathcal{A}$. Let g_2 be an unknown random generator of $\mathbb{G}_{p_2}$. $\mathcal{B}$ implicitly sets $\hat{\boldsymbol{h}} \bmod p_2 = \boldsymbol{h} \bmod p_2$. This is properly distributed $\hat{\boldsymbol{h}}$; in particular, it is independent from $\boldsymbol{h} \bmod p_1$ due to the Chinese Remainder Theorem.

Phase 1. When $\mathcal{A}$ makes the j-th private key query for $X_j \in \mathbb{X}$, $\mathcal{B}$ does as follows.

[**Case** $j < k$] In this case, the algorithm $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$ and picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}'_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot (W_2W_3)^{\boldsymbol{k}(\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{R}_3.$$

It is a properly distributed type-3 semi-functional key of Equation (6) with $\hat{\alpha}_j = w_2\hat{\alpha}'_j \pmod{p_2}$, when we write $W_2W_3 = g_2^{w_2}g_3^{w_3}$.

[**Case** $j = k$] $\mathcal{B}$ runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$ and picks $\boldsymbol{r}'_j, \hat{\boldsymbol{r}}'_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and construct a key as

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j, \boldsymbol{h})} \cdot (T)^{\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} \cdot \boldsymbol{R}_3.$$

We claim that it is a properly distributed key for normal type or semi-functional type-1. To see this, first we observe that its $\mathbb{G}_{p_1}$ component is

$$g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j, \boldsymbol{h}) + t_1\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j, \boldsymbol{h})},$$

where we write $T = g_1^{t_1}g_2^{t_2}g_3^{t_3}$. This is properly distributed as in Equation (2),(4) with $\boldsymbol{r}_j = \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j \pmod{p_1}$. We note that the linearity property is used here. If $T \in \mathbb{G}_{p_1p_3}$, then it has no $\mathbb{G}_{p_2}$ component hence is a normal key. If $T \in \mathbb{G}_{p_1p_2p_3}$, then this is a type-1 semi-functional key of which the $\mathbb{G}_{p_2}$ component is

$$g_2^{t_2\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = g_2^{\boldsymbol{k}(0, t_2\hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = g_2^{\boldsymbol{k}(0, t_2\hat{\boldsymbol{r}}'_j, \hat{\boldsymbol{h}})},$$

which is properly distributed as in Equation (4) with $\hat{\boldsymbol{r}}_j = t_2\hat{\boldsymbol{r}}'_j \pmod{p_2}$.

[**Case** $j > k$] The algorithm $\mathcal{B}$ generates a normal key as in the construction. This can be done since it knows the master key $\mathsf{msk} = \alpha$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^{\star}$. Then, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0, 1\}$, runs $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^{\star})$, picks $\boldsymbol{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge $(\boldsymbol{C}, C_0)$ as

$$\boldsymbol{C} = (Z_1Z_2)^{\boldsymbol{c}(\boldsymbol{s}', \boldsymbol{h})}, \qquad C_0 = (e(Z_1Z_2, g_1)^{\alpha})^{s'}M_b.$$

This is a properly distributed semi-functional ciphertext of Equation (3) with $\boldsymbol{s} = z_1\boldsymbol{s}' \pmod{p_1}$ and $\hat{\boldsymbol{s}} = z_2\boldsymbol{s}' \pmod{p_2}$, where we write $Z_1Z_2 = g_1^{z_1}g_2^{z_2}$.

Phase 2. For each query in this phase, $\mathcal{B}$ generates a normal key as in the construction.

Guess. The algorithm $\mathcal{B}$ has properly simulated game $\mathsf{G}_{k-1,3}$ if $T \in \mathbb{G}_{p_1p_3}$ and $\mathsf{G}_{k,1}$ if $T \in \mathbb{G}_{p_1p_2p_3}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break the assumption 2. □

Lemma 30. *For $k \in [1, q_1]$, if an adversary $\mathcal{A}$ can distinguish $\mathsf{G}_{k,1}$ from $\mathsf{G}_{k,2}$, we can build an algorithm $\mathcal{B}$ that breaks the co-selective master-key hiding security of the underlying pair encoding system P. Indeed, $|\mathsf{G}_{k,1}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{k,2}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}}^{\mathsf{CMH}}(\lambda)$.*

Proof. Suppose we have an adversary $\mathcal{A}$ that can distinguish between $\mathsf{G}_{k,1}$ and $\mathsf{G}_{k,2}$ with non-negligible probability. We construct a simulator $\mathcal{B}$ that would win the co-selective game for master-key hiding for P by simulating $\mathsf{G}_{k,1}$ or $\mathsf{G}_{k,2}$ for $\mathcal{A}$. In the co-selective game for $\mathcal{B}$, $\mathcal{B}$ is given parameter $g_1 \in \mathbb{G}_{p_1}, g_2 \in \mathbb{G}_{p_2}, g_3 \in \mathbb{G}_{p_3}$ from its challenger.

Setup. Algorithm $\mathcal{B}$ generates the public key as in the real construction but using the given g_1, g_3. It picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha \xleftarrow{\$} \mathbb{Z}_N$ and sets $\mathsf{PK} = (g_1, e(g_1, g_1)^\alpha, g_3, g_1^{\boldsymbol{h}})$. It sends PK to the adversary $\mathcal{A}$.

Phase 1. When $\mathcal{A}$ makes the j-th private key query for $X_j \in \mathbb{X}$, $\mathcal{B}$ does as follows.

[**Case** $j < k$] In this case, the algorithm $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$ and picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha}_j, \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{R}_3,$$

This is a properly distributed type-3 semi-functional key of Equation (6).

[**Case** $j = k$] The algorithm $\mathcal{B}$ makes a key query X_k to its challenger and receives $\boldsymbol{T} = g_2^{\boldsymbol{k}_{X_k}(\beta, \hat{\boldsymbol{r}}, \hat{\boldsymbol{h}})}$. This is the challenge for $\mathcal{B}$ to guess if $\beta = 0$ or β is randomly chosen in $\mathbb{Z}_{p_2}$. The crucial point here is that up to this point the semi-functional parameter has not been defined since it is not necessary for all the previous keys. This feature is known as *delayed parameter*. $\mathcal{B}$ implicitly defines the semi-functional parameter to $\hat{\boldsymbol{h}}$, which is defined in $\boldsymbol{T}$. This is done by answering back to $\mathcal{A}$ the key for X_k by using $\boldsymbol{T}$. More precisely, $\mathcal{B}$ runs $(\boldsymbol{k}_{X_k}; m_2) \leftarrow \mathsf{Enc1}(X_k)$, picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}_{X_k}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot \boldsymbol{T} \cdot \boldsymbol{R}_3.$$

This is a properly distributed semi-functional key of type-1 if $\beta = 0$ or type-2 if β is random.

[**Case** $j > k$] The algorithm $\mathcal{B}$ generates a normal key as in the construction. This can be done since it knows the master key $\mathsf{msk} = \alpha$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$ such that $R(X_j, Y^\star) = 0$ for all $j \in [1, q_1]$. $\mathcal{B}$ then makes a ciphertext query for $Y^\star$ to its challenger and receives back $\boldsymbol{D} = g_2^{\boldsymbol{c}_{Y^\star}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{h}})}$. This query can be made since $R(X_k, Y^\star) = 0$. Then, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0, 1\}$, runs $(\boldsymbol{c}_{Y^\star}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $\boldsymbol{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge ciphertext as

$$\boldsymbol{C} = g_1^{\boldsymbol{c}_{Y^\star}(\boldsymbol{s}, \boldsymbol{h})} \cdot \boldsymbol{D}, \qquad\qquad C_0 = (e(g_1, g_1)^\alpha)^s M_b.$$

This is a properly distributed semi-functional ciphertext of Equation (3).

Phase 2. For each query in this phase, $\mathcal{B}$ generates a normal key as in the construction.

Guess. The algorithm $\mathcal{B}$ has properly simulated $\mathsf{G}_{k,1}$ if $\beta = 0$, and $\mathsf{G}_{k,2}$ if β is random. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to guess β. □

Lemma 31. *For $k \in [1, q_1]$, if an adversary $\mathcal{A}$ can distinguish $\mathsf{G}_{k,2}$ from $\mathsf{G}_{k,3}$, we can build an algorithm $\mathcal{B}$ that breaks Assumption 2. Indeed, $|\mathsf{G}_{k,2}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{k,3}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD2}}(\lambda)$.*

Proof. Our algorithm $\mathcal{B}$ takes in a problem instance $(g_1, Z_1Z_2, Z_3, W_2W_3, T)$ for Assumption 2. Its task will be to decide whether $T \in \mathbb{G}_{p_1p_2p_3}$ or $T \in \mathbb{G}_{p_1p_3}$.

Setup. Algorithm $\mathcal{B}$ picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha, \xleftarrow{\$} \mathbb{Z}_N$, and prepares the public key $\mathsf{PK} = (g_1, e(g_1, g_1)^{\alpha}, Z_3, g_1^{\boldsymbol{h}})$ as usual. It sends PK to the adversary $\mathcal{A}$. Let g_2 be an unknown random generator of $\mathbb{G}_{p_2}$. $\mathcal{B}$ implicitly sets $\hat{\boldsymbol{h}}$ mod $p_2 = \boldsymbol{h}$ mod p_2. This is properly distributed $\hat{\boldsymbol{h}}$; in particular, it is independent from $\boldsymbol{h}$ mod p_1 due to the Chinese Remainder Theorem.

Phase 1. When $\mathcal{A}$ makes the j-th private key query, $\mathcal{B}$ does as follows.

[**Case** $j < k$] In this case, the algorithm $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}'_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot (W_2W_3)^{\boldsymbol{k}(\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{R}_3.$$

It is a properly distributed type-3 semi-functional key of Equation (6) with $\hat{\alpha}_j = w_2\hat{\alpha}'_j \pmod{p_2}$, when we write $W_2W_3 = g_2^{w_2}g_3^{w_3}$.

[**Case** $j = k$] $\mathcal{B}$ runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}'_j, \hat{\boldsymbol{r}}'_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}'_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and constructs a key as

$$\boldsymbol{K} = {g_1}^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j, \boldsymbol{h})} \cdot (W_2W_3)^{\boldsymbol{k}(\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0})} \cdot (T)^{\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} \cdot \boldsymbol{R}_3.$$

We claim that it is a properly distributed key for semi-functional type-2 or type-3. To see this, first we observe that its $\mathbb{G}_{p_1}$ component is

$${g_1}^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j, \boldsymbol{h}) + t_1\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = {g_1}^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j, \boldsymbol{h})},$$

where we write $T = g_1^{t_1}g_2^{t_2}g_3^{t_3}$. This is properly distributed as in Equation (5),(6) with $\boldsymbol{r}_j = \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j \pmod{p_1}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then this is a type-2 semi-functional key of which the $\mathbb{G}_{p_2}$ component is

$${g_2}^{w_2\boldsymbol{k}(\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0}) + t_2\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = {g_2}^{\boldsymbol{k}(w_2\hat{\alpha}'_j, t_2\hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = {g_2}^{\boldsymbol{k}(w_2\hat{\alpha}'_j, t_2\hat{\boldsymbol{r}}'_j, \hat{\boldsymbol{h}})},$$

which is properly distributed as in Equation (5) with $\hat{\alpha}_j = w_2\hat{\alpha}'_j \pmod{p_2}$, $\hat{\boldsymbol{r}}_j = t_2\hat{\boldsymbol{r}}'_j \pmod{p_2}$. If $T \in \mathbb{G}_{p_1p_3}$, then this is a type-3 semi-functional key of which the $\mathbb{G}_{p_2}$ component is

$${g_2}^{w_2\boldsymbol{k}(\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0})} = {g_2}^{\boldsymbol{k}(w_2\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0})},$$

which is properly distributed as in Equation (6).

[**Case** $j > k$] The algorithm $\mathcal{B}$ generates a normal key as in the construction. This can be done since it knows the master key $\mathsf{msk} = \alpha$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$. Then, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0, 1\}$, runs $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $\boldsymbol{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge $(\boldsymbol{C}, C_0)$ as

$$\boldsymbol{C} = (Z_1Z_2)^{\boldsymbol{c}(\boldsymbol{s}', \boldsymbol{h})}, \qquad\qquad C_0 = (e(Z_1Z_2, g_1)^{\alpha})^{s'}M_b.$$

This is a properly distributed semi-functional ciphertext of Equation (3) with $\boldsymbol{s} = z_1\boldsymbol{s}' \pmod{p_1}$ and $\hat{\boldsymbol{s}} = z_2\boldsymbol{s}' \pmod{p_2}$, where we write $Z_1Z_2 = g_1^{z_1}g_2^{z_2}$.

Phase 2. For each query, $\mathcal{B}$ generates a normal key as in the construction.

Guess. The algorithm $\mathcal{B}$ has properly simulated $\mathsf{G}_{k,3}$ if $T \in \mathbb{G}_{p_1p_3}$ and $\mathsf{G}_{k,2}$ if $T \in \mathbb{G}_{p_1p_2p_3}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break the assumption 2. □

Lemma 32. *If an adversary $\mathcal{A}$ can distinguish $\mathsf{G}_{q_1,3}$ from G_{q_1+1}, we can build an algorithm $\mathcal{B}$ that breaks Assumption 2. Indeed, we have $|\mathsf{G}_{q_1,3}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{q_1+1}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD2}}(\lambda)$.*

Proof. Our algorithm $\mathcal{B}$ takes in a problem instance $(g_1, Z_1Z_2, Z_3, W_2W_3, T)$ for Assumption 2. Its task will be to decide whether $T \in \mathbb{G}_{p_1p_2p_3}$ or $T \in \mathbb{G}_{p_1p_3}$.

Setup. Algorithm $\mathcal{B}$ picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha, \xleftarrow{\$} \mathbb{Z}_N$, and prepares the public key $\mathsf{PK} = (g_1, e(g_1,g_1)^{\alpha}, Z_3, g_1^{\boldsymbol{h}})$ as usual. It sends PK to the adversary $\mathcal{A}$. Let g_2 be an unknown random generator of $\mathbb{G}_{p_2}$. $\mathcal{B}$ implicitly sets $\hat{\boldsymbol{h}}$ mod $p_2 = \boldsymbol{h}$ mod p_2. This is properly distributed $\hat{\boldsymbol{h}}$; in particular, it is independent from $\boldsymbol{h}$ mod p_1 due to the Chinese Remainder Theorem.

Phase 1. When $\mathcal{A}$ makes the j-th private key query ($j \le q_1$), $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}'_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r}_j,\boldsymbol{h})} \cdot (W_2W_3)^{\boldsymbol{k}(\hat{\alpha}'_j,\boldsymbol{0},\boldsymbol{0})} \cdot \boldsymbol{R}_3.$$

It is a properly distributed type-3 semi-functional key of Equation (6) with $\hat{\alpha}_j = w_2\hat{\alpha}'_j \pmod{p_2}$, when we write $W_2W_3 = g_2^{w_2}g_3^{w_3}$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$. Then, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0,1\}$, runs $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $\boldsymbol{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge $(\boldsymbol{C}, C_0)$ as

$$\boldsymbol{C} = (Z_1Z_2)^{\boldsymbol{c}(\boldsymbol{s}',\boldsymbol{h})}, \qquad C_0 = (e(Z_1Z_2, g_1)^{\alpha})^{s'}M_b.$$

This is a properly distributed semi-functional ciphertext of Equation (3) with $\boldsymbol{s} = z_1\boldsymbol{s}' \pmod{p_1}$ and $\hat{\boldsymbol{s}} = z_2\boldsymbol{s}' \pmod{p_2}$, where we write $Z_1Z_2 = g_1^{z_1}g_2^{z_2}$.

Phase 2. When $\mathcal{A}$ makes the j-th private key query ($j > q_1$), the algorithm $\mathcal{B}$ runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$ and picks $\boldsymbol{r}'_j, \hat{\boldsymbol{r}}'_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and construct a key as

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r}'_j,\boldsymbol{h})} \cdot (T)^{\boldsymbol{k}(0,\hat{\boldsymbol{r}}'_j,\boldsymbol{h})} \cdot \boldsymbol{R}_3.$$

We claim that it is a properly distributed key for normal type or semi-functional type-1. To see this, first we observe that its $\mathbb{G}_{p_1}$ component is

$$g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r}'_j,\boldsymbol{h})+t_1\boldsymbol{k}(0,\hat{\boldsymbol{r}}'_j,\boldsymbol{h})} = g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r}'_j+t_1\hat{\boldsymbol{r}}'_j,\boldsymbol{h})},$$

where we write $T = g_1^{t_1}g_2^{t_2}g_3^{t_3}$. This is properly distributed as in Equation (2),(4) with $\boldsymbol{r}_j = \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j \pmod{p_1}$. We note that the linearity property is used here. If $T \in \mathbb{G}_{p_1p_3}$, then it has no $\mathbb{G}_{p_2}$ component hence is a normal key. If $T \in \mathbb{G}_{p_1p_2p_3}$, then this is a type-1 semi-functional key of which the $\mathbb{G}_{p_2}$ component is

$$g_2^{t_2\boldsymbol{k}(0,\hat{\boldsymbol{r}}'_j,\boldsymbol{h})} = g_2^{\boldsymbol{k}(0,t_2\hat{\boldsymbol{r}}'_j,\boldsymbol{h})} = g_2^{\boldsymbol{k}(0,t_2\hat{\boldsymbol{r}}'_j,\hat{\boldsymbol{h}})},$$

which is properly distributed as in Equation (4) with $\hat{\boldsymbol{r}}_j = t_2\hat{\boldsymbol{r}}'_j \pmod{p_2}$.

Guess. The algorithm $\mathcal{B}$ has properly simulated $\mathsf{G}_{q_1,3}$ if $T \in \mathbb{G}_{p_1p_3}$ and G_{q_1+1} if $T \in \mathbb{G}_{p_1p_2p_3}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break the assumption 2. □

Lemma 33. *If an adversary $\mathcal{A}$ can distinguish G_{q_1+1} from G_{q_1+2}, we can build an algorithm $\mathcal{B}$ that breaks the selective master-key hiding security of the underlying pair encoding system P. Indeed, $|\mathsf{G}_{q_1+1}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{q_1+2}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SMH}}(\lambda)$.*

Proof. Suppose we have an adversary $\mathcal{A}$ that can distinguish between G_{q_1+1} and G_{q_1+2} with non-negligible probability. We construct a simulator $\mathcal{B}$ that would win the selective game for master-key hiding for P by simulating G_{q_1+1} or G_{q_1+2} for $\mathcal{A}$. In the selective game for $\mathcal{B}$, $\mathcal{B}$ is given parameter $g_1 \in \mathbb{G}_{p_1}, g_2 \in \mathbb{G}_{p_2}, g_3 \in \mathbb{G}_{p_3}$ from its challenger.

Setup. Algorithm $\mathcal{B}$ generates the public key as in the real construction but using the given g_1, g_3. It picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha \xleftarrow{\$} \mathbb{Z}_N$, and sets $\mathsf{PK} = (g_1, e(g_1, g_1)^\alpha, g_3, g_1^{\boldsymbol{h}})$. It sends PK to the adversary $\mathcal{A}$.

Phase 1. When $\mathcal{A}$ makes the j-th private key query for $X_j \in \mathbb{X}$, $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$ and picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha}_j, \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{R}_3,$$

This is a properly distributed type-3 semi-functional key of Equation (6).

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$. $\mathcal{B}$ then makes a ciphertext query for $Y^\star$ to its challenger and receives back $\boldsymbol{D} = g_2^{\boldsymbol{c}_{Y^\star}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{h}})}$. The crucial point here is that up to this point the semi-functional parameter have not been defined since it is not necessary for all the previous keys. $\mathcal{B}$ implicitly defines the semi-functional parameter to $\hat{\boldsymbol{h}}$, which is defined in $\boldsymbol{D}$. This is done by answering back to $\mathcal{A}$ the ciphertext for $Y^\star$ by using $\boldsymbol{D}$. More precisely, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0,1\}$, runs $(\boldsymbol{c}_{Y^\star}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $\boldsymbol{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge ciphertext as

$$\boldsymbol{C} = g_1^{\boldsymbol{c}_{Y^\star}(\boldsymbol{s}, \boldsymbol{h})} \cdot \boldsymbol{D}, \qquad\qquad C_0 = (e(g_1, g_1)^\alpha)^s M_b.$$

This is a properly distributed semi-functional ciphertext of Equation (3).

Phase 2. When $\mathcal{A}$ makes the j-th private key query for $X_j \in \mathbb{X}$ such that $R(X_j, Y^\star) = 0$, $\mathcal{B}$ makes a key query X_j to its challenger and receives back $\boldsymbol{T}_j = g_2^{\boldsymbol{k}_{X_j}(\beta, \hat{\boldsymbol{r}}_j, \hat{\boldsymbol{h}})}$. This query can be made since $R(X_j, Y^\star) = 0$. We note that the task for $\mathcal{B}$ to guess if $\beta = 0$ or β is randomly chosen in $\mathbb{Z}_{p_2}$. $\mathcal{B}$ then runs $(\boldsymbol{k}_{X_j}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, randomly chooses $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes the key as.

$$\boldsymbol{K} = g_1^{\boldsymbol{k}_{X_j}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot \boldsymbol{T}_j \cdot \boldsymbol{R}_3.$$

This is a properly distributed semi-functional key of type-1 if $\beta = 0$ or type-2 if β is random.

Guess. We note that all the keys in phase 2 have the same $\hat{\alpha}(= \beta)$ in the $\mathbb{G}_{p_2}$ component, but this exactly corresponds to our definition for game G_{q_1+1} or G_{q_1+2}. Indeed, the algorithm $\mathcal{B}$ has properly simulated G_{q_1+1} if $\beta = 0$, and G_{q_1+2} if β is random. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to guess β. □

Lemma 34. *If an adversary $\mathcal{A}$ can distinguish G_{q_1+2} from G_{q_1+3}, we can build an algorithm $\mathcal{B}$ that breaks Assumption 2. Indeed, we have $|\mathsf{G}_{q_1+2}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{q_1+3}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD2}}(\lambda)$.*

Proof. Our algorithm $\mathcal{B}$ takes in a problem instance $(g_1, Z_1Z_2, Z_3, W_2W_3, T)$ for Assumption 2. Its task will be to decide whether $T \in \mathbb{G}_{p_1p_2p_3}$ or $T \in \mathbb{G}_{p_1p_3}$.

Setup. Algorithm $\mathcal{B}$ picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$, $\alpha, \xleftarrow{\$} \mathbb{Z}_N$, and prepares the public key $\mathsf{PK} = (g_1, e(g_1, g_1)^\alpha, Z_3, g_1^{\boldsymbol{h}})$ as usual. It sends PK to the adversary $\mathcal{A}$. Let g_2 be an unknown random generator of $\mathbb{G}_{p_2}$. $\mathcal{B}$

implicitly sets $\hat{\boldsymbol{h}} \bmod p_2 = \boldsymbol{h} \bmod p_2$. This is properly distributed $\hat{\boldsymbol{h}}$; in particular, it is independent from $\boldsymbol{h}$ mod p_1 due to the Chinese Remainder Theorem.

Phase 1. When $\mathcal{A}$ makes the j-th private key query for X_j, the algorithm $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\hat{\alpha}'_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = g_1^{\boldsymbol{k}(\alpha, \boldsymbol{r}_j, \boldsymbol{h})} \cdot (W_2W_3)^{\boldsymbol{k}(\hat{\alpha}'_j, \boldsymbol{0}, \boldsymbol{0})} \cdot \boldsymbol{R}_3.$$

It is a properly distributed type-3 semi-functional key of Equation (6) with $\hat{\alpha}_j = w_2\hat{\alpha}'_j \pmod{p_2}$, when we write $W_2W_3 = g_2^{w_2}g_3^{w_3}$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$. Then, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0,1\}$, runs $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $\boldsymbol{s}' = (s', s'_1, \ldots, s'_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$, and forms the challenge $(\boldsymbol{C}, C_0)$ as

$$\boldsymbol{C} = (Z_1Z_2)^{\boldsymbol{c}(\boldsymbol{s}', \boldsymbol{h})}, \qquad C_0 = (e(Z_1Z_2, g_1)^{\alpha})^{s'} M_b.$$

This is a properly distributed semi-functional ciphertext of Equation (3) with $\boldsymbol{s} = z_1\boldsymbol{s}' \pmod{p_1}$ and $\hat{\boldsymbol{s}} = z_2\boldsymbol{s}' \pmod{p_2}$, where we write $Z_1Z_2 = g_1^{z_1}g_2^{z_2}$.

Phase 2. At the beginning of this phase, $\mathcal{B}$ picks $\hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. When $\mathcal{A}$ makes the j-th private key query for X_j, the algorithm $\mathcal{B}$ runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}'_j, \hat{\boldsymbol{r}}'_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and constructs a key as

$$\boldsymbol{K} = g_1{}^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j, \boldsymbol{h})} \cdot (W_2W_3)^{\boldsymbol{k}(\hat{\alpha}', \boldsymbol{0}, \boldsymbol{0})} \cdot (T)^{\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} \cdot \boldsymbol{R}_3.$$

We claim that it is a properly distributed key for semi-functional type-2 or type-3. To see this, first we observe that its $\mathbb{G}_{p_1}$ component is

$$g_1{}^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j, \boldsymbol{h}) + t_1\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = g_1{}^{\boldsymbol{k}(\alpha, \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j, \boldsymbol{h})},$$

where we write $T = g_1^{t_1}g_2^{t_2}g_3^{t_3}$. This is properly distributed as in Equation (5),(6) with $\boldsymbol{r}_j = \boldsymbol{r}'_j + t_1\hat{\boldsymbol{r}}'_j \pmod{p_1}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then this is a type-2 semi-functional key of which the $\mathbb{G}_{p_2}$ component is

$$g_2{}^{w_2\boldsymbol{k}(\hat{\alpha}', \boldsymbol{0}, \boldsymbol{0}) + t_2\boldsymbol{k}(0, \hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = g_2{}^{\boldsymbol{k}(w_2\hat{\alpha}', t_2\hat{\boldsymbol{r}}'_j, \boldsymbol{h})} = g_2{}^{\boldsymbol{k}(w_2\hat{\alpha}', t_2\hat{\boldsymbol{r}}'_j, \hat{\boldsymbol{h}})},$$

which is properly distributed as in Equation (5) with $\hat{\alpha}_j = w_2\hat{\alpha}' \pmod{p_2}$, $\hat{\boldsymbol{r}}_j = t_2\hat{\boldsymbol{r}}'_j \pmod{p_2}$. If $T \in \mathbb{G}_{p_1p_3}$, then this is a type-3 semi-functional key of which the $\mathbb{G}_{p_2}$ component is

$$g_2{}^{w_2\boldsymbol{k}(\hat{\alpha}', \boldsymbol{0}, \boldsymbol{0})} = g_2{}^{\boldsymbol{k}(w_2\hat{\alpha}', \boldsymbol{0}, \boldsymbol{0})},$$

which is properly distributed as in Equation (6).

Guess. We note that all the keys in phase 2 have the same β in the $\mathbb{G}_{p_2}$ component, but this exactly corresponds to our definition for game G_{q_1+2} or G_{q_1+3}. Indeed, the algorithm $\mathcal{B}$ has properly simulated game G_{q_1+3} if $T \in \mathbb{G}_{p_1p_3}$ and G_{q_1+2} if $T \in \mathbb{G}_{p_1p_2p_3}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break the assumption 2. □

Lemma 35. *If an adversary $\mathcal{A}$ can distinguish G_{q_1+3} from $\mathsf{G}_{\mathsf{final}}$, we can build an algorithm $\mathcal{B}$ that breaks Assumption 3. Indeed,* $|\mathsf{G}_{q_1+3}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda) - \mathsf{G}_{\mathsf{final}}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FE}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD3}}(\lambda)$.

Proof. Our algorithm $\mathcal{B}$ takes as input a problem instance $(g_1, g_2, Z_3, g_1^s W_2, g_1^\alpha Y_2, T)$ for Assumption 3. Its task will be to decide whether $T = e(g,g)^{\alpha s}$ or T is randomly chosen from $\mathbb{G}_T$.

Setup. Algorithm $\mathcal{B}$ picks $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$. It then computes $e(g_1, g_1^\alpha Y_2) = e(g_1, g_1)^\alpha$ and prepares the public key $\mathsf{PK} = \big(g_1, e(g_1, g_1)^\alpha, Z_3, g_1^{\boldsymbol{h}}\big)$. It sends PK to the adversary $\mathcal{A}$. Since all the keys in game G_{q_1+3} and $\mathsf{G}_{\text{final}}$ will be semi-functional of type-3, we do not need to define $\hat{\boldsymbol{h}}$.

Phase 1. When $\mathcal{A}$ makes a private key query for X_j, the algorithm $\mathcal{B}$ creates a type-3 semi-functional key. To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}, \hat{\alpha}'_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = (g_1^\alpha Y_2)^{\boldsymbol{k}(1,\boldsymbol{0},\boldsymbol{0})} \cdot g_1{}^{\boldsymbol{k}(0,\boldsymbol{r}_j,\boldsymbol{h})} \cdot g_2{}^{\boldsymbol{k}(\hat{\alpha}'_j,\boldsymbol{0},\boldsymbol{0})} \cdot \boldsymbol{R}_3.$$

It is a properly distributed type-3 semi-functional key of Equation (6) with $\hat{\alpha}_j = y_2 + \hat{\alpha}'_j \pmod{p_2}$, when we write $Y_2 = g_2^{y_2}$.

Challenge. In the challenge phase, the adversary $\mathcal{A}$ outputs messages $M_0, M_1 \in \mathbb{G}_T$ along with her target $Y^\star$. Then, $\mathcal{B}$ flips a coin $b \xleftarrow{\$} \{0,1\}$, runs $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$, picks $s_1, \ldots, s_{w_2} \xleftarrow{\$} \mathbb{Z}_N$, defines $\boldsymbol{s}' = (1, s_1, \ldots, s_{w_2})$, and forms the challenge ciphertext as

$$\boldsymbol{C} = (g_1^s W_2)^{\boldsymbol{c}(\boldsymbol{s}',\boldsymbol{h})}, \qquad\qquad C_0 = TM_b.$$

The $\boldsymbol{C}$ term is a properly distributed as in a semi-functional ciphertext of Equation (3) with $\boldsymbol{s} = s\boldsymbol{s}' \pmod{p_1}$, $\hat{\boldsymbol{s}} = w_2 \boldsymbol{s}' \pmod{p_2}$, where we write $W_2 = g_2^{w_2}$.

Phase 2. At the beginning of this phase, $\mathcal{B}$ picks $\hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$. When $\mathcal{A}$ makes a private key query for X_j, the algorithm $\mathcal{B}$ creates a type-3 semi-functional key, with all the keys share the same $\hat{\alpha}$ (as per the definition of game G_{q_1+3} or $\mathsf{G}_{\text{final}}$). To do so, it runs $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_j)$, picks $\boldsymbol{r}_j \xleftarrow{\$} \mathbb{Z}_N^{m_2}$, $\boldsymbol{R}_3 \xleftarrow{\$} \mathbb{G}_{p_3}^{m_1}$, and computes

$$\boldsymbol{K} = (g_1^\alpha Y_2)^{\boldsymbol{k}(1,\boldsymbol{0},\boldsymbol{0})} \cdot (g_1)^{\boldsymbol{k}(0,\boldsymbol{r}_j,\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha}',\boldsymbol{0},\boldsymbol{0})} \cdot \boldsymbol{R}_3.$$

It is a properly distributed type-3 semi-functional key of Equation (6) with $\hat{\alpha} = y_2 + \hat{\alpha}' \pmod{p_2}$, when we write $Y_2 = g_2^{y_2}$.

Guess. The algorithm $\mathcal{B}$ has properly simulated G_{q_1+3} if $T = e(g,g)^{\alpha s}$ and $\mathsf{G}_{\text{final}}$ if T is randomly chosen. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break the assumption 3. □

A.2 Proof for Theorem with Normal Reduction

In this section, we describe the security proof of Theorem 2. All the indistinguishability proofs in the sequence are done in exactly the same way as those for Theorem 1 except only the indistinguishability between game $\mathsf{G}_{k,1}$ and $\mathsf{G}_{k,2}$ (for all $k \in [1, q_{\text{all}}]$), which we show below.

Lemma 36. *Suppose that the underlying pair encoding is perfectly master-key hiding. For any $k \in [1, q_1 + q_2]$, $\mathsf{G}_{k,1}$ and $\mathsf{G}_{k,2}$ are indistinguishable in the information theoretic sense.*

Proof. Let $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X_k)$ and $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star)$. Since $R_{p_2}(X, Y) = 0$, from the perfect security of encoding scheme we have that the following two distributions are identical:

$$\Big\{\boldsymbol{c}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{h}}),\ \boldsymbol{k}(0, \hat{\boldsymbol{r}}, \hat{\boldsymbol{h}})\Big\} \qquad \text{and} \qquad \Big\{\boldsymbol{c}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{h}}),\ \boldsymbol{k}(\hat{\alpha}, \hat{\boldsymbol{r}}, \hat{\boldsymbol{h}})\Big\},$$

where the probability is taken over $\hat{h}_1, \ldots, \hat{h}_n, \hat{\alpha}, \hat{r}_1, \ldots, \hat{r}_{m_2}, \hat{s}, \hat{s}_1, \ldots, \hat{s}_{w_2} \xleftarrow{\$} \mathbb{Z}_{p_2}$. Therefore, the following two distribution are also identical:

$$\left\{ g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})} \cdot g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})},\ \ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \cdot \boldsymbol{R}_3 \right\} \quad \text{and} \quad \left\{ g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})} \cdot g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})},\ \ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \cdot \boldsymbol{R}_3 \right\}$$

But this is the only difference between $\mathsf{G}_{k,1}$ from $\mathsf{G}_{k,2}$. Moreover, in both games, $\alpha, \hat{\boldsymbol{h}}$ do not appear in all the other keys and $\hat{\boldsymbol{r}}$ is randomly chosen for each key. Therefore, $\mathsf{G}_{k,1}$ and $\mathsf{G}_{k,2}$ are identically distributed. This concludes the proof. □

B Proof for Functional Encryption for Regular Languages

B.1 Disproving Perfect Security

In this section, we prove that Scheme 3, our pair encoding for regular languages functionality, is not perfectly master-key hiding (Lemma 6). This is similar to Lemma 5 for Waters' scheme.

Proof. We consider $M, \boldsymbol{w}$ such that M, upon input $\boldsymbol{w}$, transits the initial state through the accepted state q_{n-1} at just after the j-th symbol of $\boldsymbol{w}$, but eventually the last state is not q_{n-1}. Hence $R(M, \boldsymbol{w}) = 0$. Now we show how to compute α from $\boldsymbol{k}_M(\alpha, \boldsymbol{r}, \boldsymbol{h})$ and $\boldsymbol{c}_{\boldsymbol{w}}(s, \boldsymbol{h})$. Firstly from our setting of $\boldsymbol{w}$, we can compute $s_j u_{n-1}$. This is done by mimicking the decryption mechanism but only until the j-th symbol. Now since $s_j u_{n-1} = s_j \phi_2 r$ and we know s_j, r, the term ϕ_2 is thus also exposed. From this and known terms s_ℓ, s, we can extract ϕ_1 from c_3. From $c_2 = s\eta$ and known term s, we have η. From ϕ_1, η and known terms r, u, we can extract α from k_1. □

B.2 Proof of Selective Security

In this section, we prove the selective master-key hiding security of Scheme 3, our pair encoding for regular languages functionality (Theorem 7 in §6).

Proof. Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the selective game for master-key hiding game against our pair encoding scheme 3 for regular language. We construct a simulator $\mathcal{B}$ that solves the ℓ-EDHE1 by running $\mathcal{A}$ and simulating the security game as follows.

Ciphertext Query. The selective game begins with $\mathcal{A}$ making a ciphertext query $\boldsymbol{w}^\star = (w_1^\star, \ldots, w_{\ell^\star}^\star)$. $\mathcal{B}$ takes as an input the $\ell^\star$-EDHE1 challenge $(\boldsymbol{D}, T)$. Its task is to guess whether $T = g^{abz}$ or T is a random element in $\mathbb{G}_p$. That is, if we denote $T = g^{\tau + abz}$, the task is to guess $\tau = 0$ or τ is a random element in $\mathbb{Z}_p$. $\mathcal{B}$ will implicitly define $\alpha = \tau$ by embedding T in the simulation for keys (see below).

(Programming Parameters). $\mathcal{B}$ computes the parameter by first picking $h'_0, \ldots, h'_4, \phi'_1 \xleftarrow{\$} \mathbb{Z}_p$ and computing

$$\begin{aligned} g^{h_0} &= g^{h'_0} \prod_{i \in [1,\ell^\star]} g^{a^i/d_i}, & g^{h_1} &= g^{h'_1} g^{a/f}, & g^{h_2} &= g^{h'_2}, \\ g^{h_3} &= g^{h'_3} g^{-1/f} \prod_{i \in [0,\ell^\star]} g^{(-w_i^\star a^{\ell^\star+1-i})/d_{\ell^\star+1-i}}, & g^{h_4} &= g^{h'_4} \prod_{i \in [0,\ell^\star]} g^{(a^{\ell^\star+1-i})/d_{\ell^\star+1-i}}, & g^{\phi_2} &= g^a. \end{aligned} \tag{9}$$

These computations are possible due to the availability of corresponding elements given in $\boldsymbol{D}$ from the assumption. It also *implicitly* defines

$$g^{\phi_1} = g^{\phi'_1} g^{az}, \qquad\qquad g^{\eta} = g^{z}. \tag{10}$$

These are implicit since $\mathcal{B}$ possesses neither g^{az}, g^{z}.

(Programming Randomness in Ciphertext). We now describe how $\mathcal{B}$ produce a ciphertext for $w^\star$. $\mathcal{B}$ first implicitly sets

$$s = a^{\ell^\star} c/z, \qquad\qquad \forall_{i=[0,\ell^\star]} \;\; s_i = a^i c.$$

(Cancellation in Ciphertext). From the above (implicit) definitions, the ciphertext is well defined. It can be computed due to the cancelation of unknown elements as follows.[7]

- Canceling $g^{a^{\ell^\star+1}c}$ in $g^{-s\phi_1+s_{\ell^\star}\phi_2}$ by

$$g^{-(a^{\ell^\star}c/z)(az)} \text{ from } g^{-(s)(\phi_1)}, \qquad\qquad g^{(a^{\ell^\star}c)(a)} \text{ from } g^{(s_{\ell^\star})(\phi_2)}$$

- Canceling $g^{a^i c/f}$ in $g^{s_{i-1}(h_1+h_2w_i^\star)+s_i(h_3+h_4w_i^\star)}$ for all $i \in [1, \ell^\star]$ by

$$g^{(a^{i-1}c)(a/f)} \text{ from } g^{(s_{i-1})(h_1)}, \qquad\qquad g^{(a^i c)(-1/f)} \text{ from } g^{(s_i)(h_3)}$$

- Canceling $g^{a^{\ell^\star+1}c/d_{\ell^\star+1-i}}$ in $g^{s_{i-1}(h_1+h_2w_i^\star)+s_i(h_3+h_4w_i^\star)}$ for all $i \in [1, \ell^\star]$ by

$$g^{(a^i c)(-w_i^\star a^{\ell^\star+1-i}/d_{\ell^\star+1-i})} \text{ from } g^{(s_i)(h_3)}, \qquad g^{(a^i c)(w_i^\star)(a^{\ell^\star+1-i}/d_{\ell^\star+1-i})} \text{ from } g^{(s_i)(w_i^\star)(h_4)}$$

(Simulating Ciphertext). The ciphertext is then computable from remaining terms which consist of only known elements from $\boldsymbol{D}$ as

$$g^{s} = g^{a^{\ell^\star}c/z}, \tag{11}$$

$$g^{s\eta} = g^{a^{\ell^\star}c}, \tag{12}$$

$$g^{-s\phi_1+s_{\ell^\star}\phi_2} = (g^{a^{\ell^\star}c/z})^{(-\phi_1')}, \tag{13}$$

$$\forall_{i\in[0,\ell^\star]} \qquad g^{s_i} = g^{a^i c}, \tag{14}$$

$$g^{s_0h_0} = (g^c)^{h_0'} \prod_{i\in[1,\ell^\star]} g^{a^i c/d_i}, \tag{15}$$

$$\forall_{i\in[1,\ell^\star]} \quad g^{s_{i-1}(h_1+h_2w_i^\star)+s_i(h_3+h_4w_i^\star)} = (g^{a^{i-1}c})^{(h_1'+h_2'w_i^\star)}(g^{a^i c})^{(h_3'+h_4'w_i^\star)} \cdot \prod_{\substack{j\in[0,\ell^\star] \\ \text{s.t.} j\neq i}} g^{(w_i^\star-w_j^\star)a^{\ell^\star+1-j+i}c/d_{\ell^\star+1-j}}. \tag{16}$$

(Re-randomizing Ciphertext). The simulated ciphertext is not perfectly distributed yet since the randomness $s, s_0, \dots, s_{\ell^\star}$ are still correlated. We will re-randomize $s_0, \dots, s_{\ell^\star}$ so that they are independent from s. This can be done by randomly choosing $s_0', \dots, s_{\ell^\star}' \xleftarrow{\$} \mathbb{Z}_p$ and computing a new ciphertext as

$$g^{\tilde{s}} = g^{s},$$

$$g^{\tilde{s}\eta} = g^{s\eta},$$

$$g^{-\tilde{s}\phi_1+\tilde{s}_{\ell^\star}\phi_2} = g^{-s\phi_1+s_{\ell^\star}\phi_2}(g^{\phi_2})^{s_{\ell^\star}'},$$

$$\forall_{i\in[0,\ell^\star]} \qquad g^{\tilde{s}_i} = g^{s_i}g^{s_i'},$$

$$g^{\tilde{s}_0h_0} = g^{s_0h_0}(g^{h_0})^{s_0'},$$

$$\forall_{i\in[1,\ell^\star]} \quad g^{\tilde{s}_{i-1}(h_1+h_2w_i^\star)+\tilde{s}_i(h_3+h_4w_i^\star)} = g^{s_{i-1}(h_1+h_2w_i^\star)+s_i(h_3+h_4w_i^\star)}((g^{h_1})(g^{h_2})^{w_i^\star})^{s_{i-1}'}((g^{h_3})(g^{h_4})^{w_i^\star})^{s_i'}$$

[7] We will use colored texts to identify what terms come from what terms.

where the new ciphertext has new randomness $\tilde{s}_i = s_i + s'_i$.

Key Query. The adversary $\mathcal{A}$ makes a key query for DFA $M = (Q, \mathcal{T}, q_0, q_{n-1})$. $\mathcal{B}$ creates the key for M as follows. We first describe some notation. Denote by $\boldsymbol{w}_i^\star$ the vector formed by the last $\ell^\star - i$ symbol of $\boldsymbol{w}^\star$. That is $\boldsymbol{w}_i^\star = (w_{i+1}^\star, \ldots, w_{\ell^\star}^\star)$. Hence $\boldsymbol{w}_0^\star = \boldsymbol{w}^\star$ and $\boldsymbol{w}_{\ell^\star}^\star$ is the empty string. For $q_k \in \{q_0, \ldots, q_{n-1}\} = Q$, let M_k be the same DFA as M except that the start state is set to q_k. Then, for each $q_k \in Q$ we define $V_k = \{i \in [0, \ell^\star] \mid R(M_k, \boldsymbol{w}_i^\star) = 1\}$. From this and the query restriction that $R(M, w^\star) = 0$, we have $0 \notin V_0$. Since there is only one accept state q_{n-1} (due to the WLOG condition) and this state has no out-going transition, we have $V_{n-1} = \{\ell^\star\}$. We denote $V_x^{+1} = \{i + 1 \mid i \in V_x\}$.

(Programming Randomness in Key). $\mathcal{B}$ first implicitly defines

$$r = b, \tag{17}$$

$$r_0 = b\Big(\sum_{i \in V_0} d_{\ell^\star+1-i}\Big), \tag{18}$$

$$u = 0, \tag{19}$$

$$\forall_{q_x \in Q \smallsetminus \{q_{n-1}\}} \quad u_x = \sum_{i \in V_x} a^{\ell^\star+1-i} b. \tag{20}$$

We note that, from the identity $u_{n-1} = \phi_2 r$, we have $u_{n-1} = ab = \sum_{i \in V_{n-1}} a^{\ell^\star+1-i} b$, due to $V_{n-1} = \{\ell^\star\}$. Hence Equation (20) indeed holds for all $q_x \in Q$. Finally, $\mathcal{B}$ implicitly defines

$$\forall_{t \in [1,m]} \quad r_t = b\Big(-\sum_{i \in V_{x_t}^{+1}} a^{\ell^\star+1-i} f - \sum_{i \in V_{x_t}^{+1} \smallsetminus V_{y_t}} d_{\ell^\star+1-i}/(\sigma_t - w_i^\star) + \sum_{i \in V_{y_t} \smallsetminus V_{x_t}^{+1}} d_{\ell^\star+1-i}/(\sigma_t - w_i^\star)\Big). \tag{21}$$

We note that the first sum can also be expressed as $\sum_{i \in V_x^{+1}} a^{\ell^\star+1-i} f = \sum_{i \in V_x} a^{\ell^\star+1-(i+1)} f$. One can see that r_t is well defined due to the following claim.

Claim 37. *For $(x, y, \sigma) \in \mathcal{T}$, suppose that $i \in V_x^{+1} \smallsetminus V_y$ or $i \in V_y \smallsetminus V_x^{+1}$. Then we have $\sigma \neq w_i^\star$.*

The claim indeed follows straightforwardly from the determinism of DFA M and is the same fact that is used in [38].

(Cancellation in Key). From the above (implicit) definitions, the key is well defined. It can be computed due to the cancelation of unknown elements as follows.[8]

- Canceling $g^{a^{\ell^\star+1-i}b}$ for all $i \in V_0$ in $g^{-u_0 + r_0 h_0}$ by

$$g^{(-a^{\ell^\star+1-i}b)} \text{ from } g^{(-u_0)}, \qquad g^{(bd_{\ell^\star+1-i})(a^{\ell^\star+1-i}/d_{\ell^\star+1-i})} \text{ from } g^{(r_0)(h_0)}$$

- Canceling $g^{a^{\ell^\star+1-i}b}$ for all $i \in V_{x_t}$ in $g^{u_{x_t} + r_t(h_1 + h_2 \sigma_t)}$ for all $t \in [1, m]$ by

$$g^{(a^{\ell^\star+1-i}b)} \text{ from } g^{(u_{x_t})}, \qquad g^{(-a^{\ell^\star+1-(i+1)}bf)(a/f)} \text{ from } g^{(r_t)(h_1)}$$

- Canceling $g^{a^{\ell^\star+1-i}b}$ for all $i \in V_{y_t} \cap V_{x_t}^{+1}$ in $g^{-u_{y_t} + r_t(h_3 + h_4 \sigma_t)}$ for all $t \in [1, m]$by

$$g^{(-a^{\ell^\star+1-i}b)} \text{ from } g^{(-u_{y_t})}, \qquad g^{(-a^{\ell^\star+1-i}bf)(-1/f)} \text{ from } g^{(r_t)(h_3)}$$

[8] We note that since u_{n-1} obeys the same Equation (20) as other u_x, we do not need to analyze in separate cases.

- Canceling $g^{a^{\ell^\star+1-i}b}$ for all $i \in V_{y_t} \setminus V_{x_t}^{+1}$ in $g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)}$ for all $t \in [1,m]$ by

$$g^{(-a^{\ell^\star+1-i}b)} \text{ from } g^{(-u_{y_t})}, \qquad g^{(\frac{bd_{\ell^\star+1-i}}{\sigma_t-w_i^\star})(-w_i^\star \frac{a^{\ell^\star+1-i}}{d_{\ell^\star+1-i}}+\sigma_t \frac{a^{\ell^\star+1-i}}{d_{\ell^\star+1-i}})} \text{ from } g^{(r_t)(h_3+h_4\sigma_t)}$$

- Canceling $g^{a^{\ell^\star+1-i}b}$ for all $i \in V_{x_t}^{+1} \setminus V_{y_t}$ in $g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)}$ for all $t \in [1,m]$ by

$$g^{(-a^{\ell^\star+1-i}bf)(-1/f)} \text{ from } g^{(r_t)(h_3)}, \qquad g^{(\frac{-bd_{\ell^\star+1-i}}{\sigma_t-w_i^\star})(-w_i^\star \frac{a^{\ell^\star+1-i}}{d_{\ell^\star+1-i}}+\sigma_t \frac{a^{\ell^\star+1-i}}{d_{\ell^\star+1-i}})} \text{ from } g^{(r_t)(h_3+h_4\sigma_t)}$$

The cancelation in the first case is possible since $0 \notin V_0$.

(Simulating Key). The key is then computable from remaining terms which consist of only known elements from $\boldsymbol{D}$ as

$$g^{\alpha+r\phi_1+u\eta} = T \cdot (g^b)^{\phi_1'} \tag{22}$$

$$g^u = 1 \tag{23}$$

$$g^r = g^b \tag{24}$$

$$g^{r_0} = g^b \tag{25}$$

and

$$g^{-u_0+r_0h_0} = (g^b)^{h_0'} \cdot \prod_{\substack{i\in V_0 \\ j\in[1,\ell^\star] \text{ s.t. } j\neq i}} g^{a^{\ell^\star+1-j}b\frac{d_{\ell^\star+1-i}}{d_{\ell^\star+1-j}}} \tag{26}$$

$$g^{r_t} = \Big(\prod_{i\in V_{x_t}^{+1}} g^{a^{\ell^\star+1-i}bf}\Big)^{-1} \cdot \Big(\prod_{i\in V_{x_t}^{+1}\setminus V_{y_t}} (g^{bd_{\ell^\star+1-i}})^{\frac{-1}{\sigma_t-w_i^\star}}\Big) \cdot \Big(\prod_{i\in V_{y_t}\setminus V_{x_t}^{+1}} (g^{bd_{\ell^\star+1-i}})^{\frac{1}{\sigma_t-w_i^\star}}\Big) \tag{27}$$

$$g^{u_{x_t}+r_t(h_1+h_2\sigma_t)} = (g^{r_t})^{(h_1'+h_2'\sigma_t)} \cdot \Big(\prod_{i\in V_{x_t}^{+1}\setminus V_{y_t}} (g^{abd_{\ell^\star+1-i}/f})^{\frac{-1}{\sigma_t-w_i^\star}}\Big) \cdot \Big(\prod_{i\in V_{y_t}\setminus V_{x_t}^{+1}} (g^{abd_{\ell^\star+1-i}/f})^{\frac{1}{\sigma_t-w_i^\star}}\Big) \tag{28}$$

$$\begin{aligned} g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)} = (g^{r_t})^{(h_3'+h_4'\sigma_t)} \cdot \Big(\prod_{\substack{i\in V_{x_t}^{+1} \\ j\in[0,\ell^\star]}} (g^{a^{2\ell^\star+2-i-j}bf/d_{\ell^\star+1-j}})^{(\sigma_t-w_j^\star)}\Big) \cdot \\ \Big(\prod_{i\in V_{x_t}^{+1}\setminus V_{y_t}} (g^{bd_{\ell^\star+1-i}/f})^{\frac{-1}{\sigma_t-w_i^\star}}\Big) \cdot \Big(\prod_{i\in V_{y_t}\setminus V_{x_t}^{+1}} (g^{bd_{\ell^\star+1-i}/f})^{\frac{1}{\sigma_t-w_i^\star}}\Big) \cdot \\ \Big(\prod_{\substack{i\in V_{x_t}^{+1}\setminus V_{y_t} \\ j\in[0,\ell^\star] \text{ s.t. } j\neq i}} (g^{a^{\ell^\star+1-j}b\frac{d_{\ell^\star+1-i}}{d_{\ell^\star+1-j}}})^{\frac{-1}{\sigma_t-w_i^\star}}\Big) \cdot \Big(\prod_{\substack{i\in V_{y_t}\setminus V_{x_t}^{+1} \\ j\in[0,\ell^\star] \text{ s.t. } j\neq i}} (g^{a^{\ell^\star+1-j}b\frac{d_{\ell^\star+1-i}}{d_{\ell^\star+1-j}}})^{\frac{1}{\sigma_t-w_i^\star}}\Big) \end{aligned} \tag{29}$$

T is embedded in such a way that $\tau = \alpha$, so that $\mathcal{B}$ can use the guess of $\mathcal{A}$ to guess in its own game.

(Re-randomizing Key). The simulated key is not perfectly distributed yet since their randomness are still correlated. $\mathcal{B}$ re-randomizes the key by randomly choosing $\forall_{x\in Q}\ u_x', r', r_0', \forall_{t\in\mathcal{T}}\ r_t' \xleftarrow{\$} \mathbb{Z}_p$

and computing a new key as follows.

$$\begin{aligned}
g^{\alpha+\tilde{r}\phi_1+\tilde{u}\eta} &= g^{\alpha+r\phi_1+u\eta}(g^{r'\phi'_1})(g^{a^{\ell^\star}c})^{u'} \\
g^{\tilde{u}} &= g^{u}(g^{a^{\ell^\star}c/z})^{u'}(g^{a})^{-r'} \\
g^{\tilde{r}} &= g^{r}g^{r'} \\
g^{\tilde{r}_0} &= g^{r_0}g^{r'_0} \\
g^{-\tilde{u}_0+\tilde{r}_0h_0} &= g^{-u_0+r_0h_0}g^{u'_0}(g^{h_0})^{r'_0} \\
g^{\tilde{r}_t} &= g^{r_t}g^{r'_t} \\
g^{\tilde{u}_{x_t}+\tilde{r}_t(h_1+h_2\sigma_t)} &= g^{u_{x_t}+r_t(h_1+h_2\sigma_t)}g^{u'_{x_t}}((g^{h_1})(g^{h_2})^{\sigma_t})^{r'_t} \\
g^{-\tilde{u}_{y_t}+\tilde{r}_t(h_3+h_4\sigma_t)} &= \begin{cases} g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)}g^{-u'_{y_t}}((g^{h_3})(g^{h_4})^{\sigma_t})^{r'_t} & \text{if } y_t \neq n-1 \\ g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)}(g^{\phi_2})^{r'}((g^{h_3})(g^{h_4})^{\sigma_t})^{r'_t} & \text{if } y_t = n-1 \end{cases}
\end{aligned}$$

where the new key has new randomness $\tilde{u}_x = u_x + u'_x, \tilde{r}_t = r_t + r'_t, \tilde{r}_0 = r_0 + r'_0, \tilde{r} = r + r', \tilde{u} = u + (a^{\ell^\star}c/z)u' - ar'$.

Guess. $\mathcal{A}$ will eventually output a guess if $\alpha = 0$ or random, our simulator $\mathcal{B}$ just outputs the guess that $\tau = 0$ and τ is random respectively. Hence, the advantage of $\mathcal{B}$ winning the $\ell^\star$-EDHE1 game is exactly the same as the advantage of $\mathcal{A}$ winning the selective master-key hiding game.

□

B.3 Proof of Co-Selective Security

In this section, we prove the co-selective master-key hiding security of Scheme 3, our pair encoding for regular languages functionality (Theorem 8 in §6).

Proof. Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the co-selective game for master-key hiding game against our pair encoding scheme 3 for regular language. We construct a simulator $\mathcal{B}$ that solves the (n, m)-EDHE2 by running $\mathcal{A}$ and simulating the security game as follows.

Key Query. The co-selective game begins with $\mathcal{A}$ making a key query $M^\star = (Q^\star, \mathcal{T}^\star, q_0, q_{n^\star-1})$. We parse $\mathcal{T}^\star = \{(x_1, y_1, \sigma_1^\star), \ldots, (x_{m^\star}, y_{m^\star}, \sigma_{m^\star}^\star)\}$, where we denote $m^\star = |\mathcal{T}^\star|$. $\mathcal{B}$ takes as an input the $(n^\star, m^\star)$-EDHE2 challenge $(\boldsymbol{D}, T)$. Its task is to guess whether $T = g^{abz}$ or T is a random element in $\mathbb{G}_p$. That is, if we denote $T = g^{\tau+abz}$, the task is to guess $\tau = 0$ or τ is a random element in $\mathbb{Z}_p$. $\mathcal{B}$ will implicitly define $\alpha = \tau$ by embedding T in the simulation for keys (see below).

(Programming Parameters). $\mathcal{B}$ computes the parameter by first randomly choosing $h'_0, \ldots, h'_4, \phi'_1 \xleftarrow{\$} \mathbb{Z}_p$ and computing

$$\begin{aligned}
&g^{h_0} = g^{h'_0}g^{a^{n^\star}b/d_1}, && g^{h_1} = g^{h'_1}\prod_{k\in[1,m^\star]} g^{\sigma_k^\star(a^{n^\star-x_k})/d_k^2}g^{-(a^{n^\star-x_k})b/d_k}, && g^{h_2} = g^{h'_2}\prod_{k\in[1,m^\star]} g^{-(a^{n^\star-x_k})/d_k^2}, \\
&g^{\phi_2} = g^{a}, && g^{h_3} = g^{h'_3}\prod_{k\in[1,m^\star]} g^{-\sigma_k^\star(a^{n^\star-y_k})/d_k^6}g^{(a^{n^\star-y_k})b/d_k}, && g^{h_4} = g^{h'_4}\prod_{k\in[1,m^\star]} g^{(a^{n^\star-y_k})/d_k^6}.
\end{aligned} \tag{30}$$

These computations are possible due to the availability of corresponding elements given in $\boldsymbol{D}$ from the assumption. It also *implicitly* defines

$$g^{\phi_1} = g^{\phi_1'} g^{az}, \qquad\qquad g^{\eta} = g^{z}. \tag{31}$$

These are implicit since $\mathcal{B}$ possesses neither g^{az}, g^{z}.

(Programming Randomness in Key). We now describe how $\mathcal{B}$ produce a key for $M^{\star}$. $\mathcal{B}$ first implicitly sets

$$u = 0, \quad r_0 = d_1, \quad r = b, \quad \forall_{t \in [1,m^{\star}]} \;\; r_t = d_t, \quad \forall_{q_x \in Q^{\star} \smallsetminus \{q_{n^{\star}-1}\}} \;\; u_x = (a^{n^{\star}-x})b. \tag{32}$$

Now recall that $u_{n^{\star}-1} = \phi_2 r$, hence from our programming we have $u_{n^{\star}-1} = ab$. Therefore, the above equation for u_x indeed holds for all $q_x \in Q^{\star}$.

(Cancellation in Key). From the above (implicit) definitions, the key is well defined. It can be computed due to the cancelation of unknown elements as follows.

- Canceling $g^{a^{n^{\star}}b}$ in $g^{u_0 + r_0 h_0}$ by
$$g^{(-a^{n^{\star}}b)} \text{ from } g^{(-u_0)}, \qquad\qquad g^{(d_1)(a^{n^{\star}}b/d_1)} \text{ from } g^{(r_0)(h_0)}$$
- Canceling $g^{(a^{n^{\star}-x_t})b}$ in $g^{u_{x_t} + r_t(h_1 + h_2\sigma_t)}$ by
$$g^{((a^{n^{\star}-x_t})b)} \text{ from } g^{(u_{x_t})}, \qquad\qquad g^{(d_t)(-(a^{n^{\star}-x_t})b/d_t)} \text{ from } g^{(r_t)(h_1)}$$
- Canceling $g^{(a^{n^{\star}-y_t})b}$ in $g^{-u_{y_t} + r_t(h_3 + h_4\sigma_t)}$ by
$$g^{(-(a^{n^{\star}-y_t})b)} \text{ from } g^{(-u_{y_t})}, \qquad\qquad g^{(d_t)((a^{n^{\star}-y_t})b/d_t)} \text{ from } g^{(r_t)(h_3)}$$
- Canceling $g^{(a^{n^{\star}-x_t})/d_t}$ in $g^{u_{x_t} + r_t(h_1 + h_2\sigma_t)}$ by
$$g^{(d_t)(\sigma_t^{\star}(a^{n^{\star}-x_t})/d_t^2)} \text{ from } g^{(r_t)(h_1)}, \qquad\qquad g^{(d_t)(\sigma_t^{\star})(-(a^{n^{\star}-x_t})/d_t^2)} \text{ from } g^{(r_t)(\sigma_t^{\star})(h_2)}$$
- Canceling $g^{(a^{n^{\star}-y_t})/d_t}$ in $g^{-u_{y_t} + r_t(h_3 + h_4\sigma_t)}$ by
$$g^{(d_t)(-\sigma_t^{\star}(a^{n^{\star}-y_t})/d_t^6)} \text{ from } g^{(r_t)(h_3)}, \qquad\qquad g^{(d_t)(\sigma_t^{\star})((a^{n^{\star}-y_t})/d_t^6)} \text{ from } g^{(r_t)(\sigma_t^{\star})(h_4)}$$

(Simulating Key). The key is then computed from all known elements from $\boldsymbol{D}$ as

$$g^{\alpha + r\phi_1 + u\eta} = T \cdot (g^b)^{\phi_1'} \tag{33}$$

$$g^{u} = 1 \tag{34}$$

$$g^{r} = g^{b} \tag{35}$$

$$g^{r_0} = g^{d_1} \tag{36}$$

$$g^{-u_0 + r_0 h_0} = (g^b)^{h_0'} \tag{37}$$

$$\forall_{t \in [1,m^{\star}]} \qquad g^{r_t} = g^{d_t} \tag{38}$$

$$\forall_{t \in [1,m^{\star}]} \qquad g^{u_{x_t} + r_t(h_1 + h_2\sigma_t^{\star})} = (g^{r_t})^{h_1' + h_2'\sigma_t^{\star}} \prod_{\substack{k \in [1,m^{\star}] \\ \text{s.t. } k \neq t}} (g^{(a^{n^{\star}-x_k})d_t/d_k^2})^{(\sigma_t^{\star} - \sigma_k^{\star})} g^{(a^{n^{\star}-x_k})bd_t/d_k} \tag{39}$$

$$\forall_{t \in [1,m^{\star}]} \qquad g^{-u_{y_t} + r_t(h_3 + h_4\sigma_t^{\star})} = (g^{r_t})^{h_3' + h_4'\sigma_t^{\star}} \prod_{\substack{k \in [1,m^{\star}] \\ \text{s.t. } k \neq t}} (g^{(a^{n^{\star}-y_k})d_t/d_k^6})^{(\sigma_t^{\star} - \sigma_k^{\star})} g^{(a^{n^{\star}-y_k})bd_t/d_k} \tag{40}$$

(Re-randomizing Key). The simulated key is not perfectly distributed yet since their randomness are still correlated. $\mathcal{B}$ re-randomizes the key by randomly choosing $\forall_{x\in Q}\ u'_x, r', r'_0, \forall_{t\in\mathcal{T}}\ r'_t \xleftarrow{\$} \mathbb{Z}_p$ and computing a new key as follows.

$$\begin{aligned}
g^{\alpha+\tilde{r}\phi_1+\tilde{u}\eta} &= g^{\alpha+r\phi_1+u\eta}(g^{r'_1\phi'_1})(g^{a^{n^\star-1}c})^{u'}\\
g^{\tilde{u}} &= g^{u}(g^{a^{n^\star-1}c/z})^{u'}(g^{a})^{-r'}\\
g^{\tilde{r}} &= g^{r}g^{r'}\\
g^{\tilde{r}_0} &= g^{r_0}g^{r'_0}\\
g^{-\tilde{u}_0+\tilde{r}_0h_0} &= g^{-u_0+r_0h_0}g^{u'_0}(g^{h_0})^{r'_0}\\
g^{\tilde{r}_t} &= g^{r_t}g^{r'_t}\\
g^{\tilde{u}_{x_t}+\tilde{r}_t(h_1+h_2\sigma_t)} &= g^{u_{x_t}+r_t(h_1+h_2\sigma_t)}g^{u'_{x_t}}((g^{h_1})(g^{h_2})^{\sigma_t})^{r'_t}\\
g^{-\tilde{u}_{y_t}+\tilde{r}_t(h_3+h_4\sigma_t)} &= \begin{cases} g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)}g^{-u'_{y_t}}((g^{h_3})(g^{h_4})^{\sigma_t})^{r'_t} & \text{if } y_t \neq n-1\\ g^{-u_{y_t}+r_t(h_3+h_4\sigma_t)}(g^{\phi_2})^{r'}((g^{h_3})(g^{h_4})^{\sigma_t})^{r'_t} & \text{if } y_t = n-1 \end{cases},
\end{aligned}$$

where the new key has new randomness $\tilde{u}_x = u_x + u'_x, \tilde{r}_t = r_t + r'_t, \tilde{r}_0 = r_0 + r'_0, \tilde{r} = r + r', \tilde{u} = u + (a^{n^\star-1}c/z)u' - ar'$.

Ciphertext Query The adversary $\mathcal{A}$ makes a ciphertext query for string $\boldsymbol{w} = (w_1, \ldots, w_\ell)$. $\mathcal{B}$ creates the ciphertext for $\boldsymbol{w}$ as follows. We first describe some notation. Denote by $\boldsymbol{w}_i$ the vector formed by the last $\ell - i$ symbol of $\boldsymbol{w}$. That is $\boldsymbol{w}_i = (w_{i+1}, \ldots, w_\ell)$. Hence $\boldsymbol{w}_0 = \boldsymbol{w}$ and $\boldsymbol{w}_\ell$ is the empty string. For $q_k \in \{q_0, \ldots, q_{n^\star-1}\} = Q$, let $M_k^\star$ be the same DFA as $M^\star$ except that the start state is set to q_k. Then, for each $i \in [0, \ell]$ we define $U_i = \{k \in [0, n^\star - 1] \mid R(M_k^\star, \boldsymbol{w}_i) = 1\}$. From this and the query restriction that $R(M^\star, \boldsymbol{w}) = 0$, we have $0 \notin U_0$. Due to the WLOG condition, we have $U_\ell = \{n^\star - 1\}$.

(Programming Randomness in Ciphertext). $\mathcal{B}$ implicitly defines

$$s = a^{n^\star-1}c/z,$$

$$s_0 = \Big(\sum_{j\in U_0} a^j c\Big)\Big(1 + \Big(b \sum_{\substack{k\in[1,m^\star]\\ \text{s.t. } \sigma_k^\star \neq w_1}} d_k/(\sigma_k^\star - w_1)\Big)\Big), \tag{41}$$

$$\forall_{i\in[1,\ell-1]}\ s_i = \Big(\sum_{j\in U_i} a^j c\Big)\Big(1 + \Big(b \sum_{\substack{k\in[1,m^\star]\\ \text{s.t. } \sigma_k^\star \neq w_{i+1}}} d_k/(\sigma_k^\star - w_{i+1})\Big) + \Big(b \sum_{\substack{k\in[1,m^\star]\\ \text{s.t. } \sigma_k^\star \neq w_i}} d_k^5/(\sigma_k^\star - w_i)\Big)\Big), \tag{42}$$

$$s_\ell = \Big(\sum_{j\in U_\ell} a^j c\Big)\Big(1 + \Big(b \sum_{\substack{k\in[1,m^\star]\\ \text{s.t. } \sigma_k^\star \neq w_\ell}} d_k^5/(\sigma_k^\star - w_\ell)\Big)\Big). \tag{43}$$

(Cancellation in Ciphertext). From the above (implicit) definitions, the ciphertext is well defined. It can be computed due to the cancelation of unknown elements as follows.

- Canceling $g^{a^{n^\star}c}$ in $g^{-s\phi_1+s_\ell\phi_2}$ by

$$g^{-(a^{n^\star-1}c/z)(az)} \text{ from } g^{-(s)(\phi_1)}, \qquad g^{(a^{n^\star-1}c)(a)} \text{ from } g^{(s_\ell)(\phi_2)}$$

- Canceling $g^{(a^{n^\star - x_t + j})bc/d_t}$ for t such that $\sigma_t^\star \neq w_i$ and for $j \in U_{i-1}$ in $g^{s_{i-1}(h_1+h_2w_i)+s_i(h_3+h_4w_i)}$ for all $i \in [1, \ell]$ by

$$g^{(a^j c)(-(a^{n^\star - x_t})b/d_t)} \text{ from } g^{(s_{i-1})(h_1)}, \quad g^{((a^j c)(\frac{bd_t}{\sigma_t^\star - w_i}))(\sigma_t^\star \frac{(a^{n^\star - x_t})}{d_t^2} + w_i^\star \frac{-(a^{n^\star - x_t})}{d_t^2})} \text{ from } g^{(s_{i-1})(h_1 + h_2 w_i^\star)}$$

- Canceling $g^{(a^{n^\star - y_t + j})bc/d_t}$ for t such that $\sigma_t^\star \neq w_i$ and for $j \in U_i$ in $g^{s_{i-1}(h_1+h_2w_i)+s_i(h_3+h_4w_i)}$ for all $i \in [1, \ell]$ by

$$g^{(a^j c)((a^{n^\star - y_t})b/d_t)} \text{ from } g^{(s_i)(h_3)}, \quad g^{((a^j c)(\frac{bd_t^5}{\sigma_t^\star - w_i}))(-\sigma_t^\star \frac{(a^{n^\star - y_t})}{d_t^6} + w_i^\star \frac{(a^{n^\star - y_t})}{d_t^6})} \text{ from } g^{(s_i)(h_3 + h_4 w_i^\star)}$$

- Canceling $g^{(a^{n^\star})bc/d_t}$ for t such that $\sigma_t^\star = w_i$ in $g^{s_{i-1}(h_1+h_2w_i)+s_i(h_3+h_4w_i)}$ for all $i \in [1, \ell]$ by

$$g^{(a^{x_t} c)(-(a^{n^\star - x_t})b/d_t)} \text{ from } g^{(s_{i-1})(h_1)}, \qquad g^{(a^{y_t} c)((a^{n^\star - y_t})b/d_t)} \text{ from } g^{(s_i)(h_3)}$$

The cancelation in the first case is due to $U_\ell = \{n^\star - 1\}$. The cancelation in the last case is possible due to the following claim.

Claim 38. *For $i \in [1, \ell]$, for t such that $\sigma_t^\star = w_i$, we have $x_t \in U_{i-1} \wedge y_t \in U_i$ or $x_t \notin U_{i-1} \wedge y_t \notin U_i$.*

The claim indeed follows straightforwardly from the determinism of DFA M and is the reverse way of stating Claim 37 in the previous proof.

(Simulating Ciphertext). Due to the cancellation of unknown elements, the ciphertext can be computed from all known elements from $\boldsymbol{D}$ as

$$g^s = g^{a^{n^\star - 1}c/z}, \tag{44}$$

$$g^{s\eta} = g^{a^{n^\star - 1}c}, \tag{45}$$

$$g^{-s\phi_1 + s_\ell \phi_2} = (g^{a^{n^\star - 1}c/z})^{-\phi_1'} \cdot \prod_{\substack{k \in [1, m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_\ell}} (g^{a^{n^\star} bcd_k^5})^{\frac{1}{\sigma_k^\star - w_\ell}}, \tag{46}$$

$$g^{s_0} = \prod_{j \in U_i} \Bigg(g^{a^j c} \prod_{\substack{k \in [1, m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_1}} (g^{a^j bcd_k})^{\frac{1}{\sigma_k^\star - w_1}} \Bigg), \tag{47}$$

$$\forall_{i \in [1, \ell - 1]} \qquad g^{s_i} = \prod_{j \in U_i} \Bigg(g^{a^j c} \prod_{\substack{k \in [1, m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_{i+1}}} (g^{a^j bcd_k})^{\frac{1}{\sigma_k^\star - w_{i+1}}} \prod_{\substack{k \in [1, m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_i}} (g^{a^j bcd_k^5})^{\frac{1}{\sigma_k^\star - w_i}} \Bigg), \tag{48}$$

$$g^{s_\ell} = \prod_{j \in U_i} \Bigg(g^{a^j c} \prod_{\substack{k \in [1, m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_\ell}} (g^{a^j bcd_k^5})^{\frac{1}{\sigma_k^\star - w_\ell}} \Bigg), \tag{49}$$

$$g^{s_0 h_0} = (g^{s_0})^{h_0'} \prod_{j \in U_0} \Bigg(g^{a^{n^\star + j} bc/d_1} \prod_{\substack{k \in [1, m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_1}} (g^{a^{n^\star + j} b^2 cd_k/d_1})^{\frac{1}{\sigma_k^\star - w_1}} \Bigg), \tag{50}$$

where we note that in the last term, $g^{a^{n^\star + j} bc/d_1}$ is available since $j \neq 0$ (due to $0 \notin U_0$).

The final components $g^{c_{6,i}} = g^{s_{i-1}(h_1+h_2 w_i^\star)+s_i(h_3+h_4 w_i^\star)}$ for $i \in [1,\ell]$ are computed as follows. For $i \in [2, \ell-1]$,

$$\begin{aligned}
g^{c_{6,i}} =& (g^{s_{i-1}})^{h_1'+h_2' w_i^\star} \cdot (g^{s_i})^{h_3'+h_4' w_i^\star} \cdot \\
&\cdot \Bigg(\prod_{\substack{t\in[1,m^\star] \\ j\in U_{i-1}}} (g^{(a^{n^\star - x_t + j})c/d_t^2})^{(\sigma_t^\star - w_i)} \Bigg) \cdot \Bigg(\prod_{\substack{t\in[1,m^\star] \\ j\in U_{i-1} \text{ s.t. } j\neq x_t}} g^{-(a^{n^\star - x_t + j})bc/d_t} \Bigg) \cdot \\
&\cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } k\neq t \wedge \sigma_k^\star \neq w_i \\ j\in U_{i-1}}} (g^{(a^{n^\star - x_t + j})bcd_k/d_t^2})^{\frac{\sigma_t^\star - w_i}{\sigma_k^\star - w_i}} \Bigg) \cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_i \\ j\in U_{i-1}}} (g^{(a^{n^\star - x_t + j})b^2cd_k/d_t})^{\frac{-1}{\sigma_k^\star - w_i}} \Bigg) \cdot \\
&\cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_{i-1} \\ j\in U_{i-1}}} (g^{(a^{n^\star - x_t + j})bcd_k^5/d_t^2})^{\frac{\sigma_t^\star - w_i}{\sigma_k^\star - w_{i-1}}} \Bigg) \cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_{i-1} \\ j\in U_{i-1}}} (g^{(a^{n^\star - x_t + j})b^2cd_k^5/d_t})^{\frac{-1}{\sigma_k^\star - w_{i-1}}} \Bigg) \cdot \\
&\cdot \Bigg(\prod_{\substack{t\in[1,m^\star] \\ j\in U_i}} (g^{-(a^{n^\star - y_t + j})c/d_t^6})^{(\sigma_t^\star - w_i)} \Bigg) \cdot \Bigg(\prod_{\substack{t\in[1,m^\star] \\ j\in U_i \text{ s.t. } j\neq y_t}} g^{(a^{n^\star - y_t + j})bc/d_t} \Bigg) \cdot \\
&\cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_{i+1} \\ j\in U_i}} (g^{(a^{n^\star - y_t + j})bcd_k/d_t^6})^{-\frac{\sigma_t^\star - w_i}{\sigma_k^\star - w_{i+1}}} \Bigg) \cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_{i+1} \\ j\in U_i}} (g^{(a^{n^\star - y_t + j})b^2cd_k/d_t})^{\frac{1}{\sigma_k^\star - w_{i+1}}} \Bigg) \cdot \\
&\cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } k\neq t \wedge \sigma_k^\star \neq w_i \\ j\in U_i}} (g^{(a^{n^\star - y_t + j})bcd_k^5/d_t^6})^{-\frac{\sigma_t^\star - w_i}{\sigma_k^\star - w_i}} \Bigg) \cdot \Bigg(\prod_{\substack{k,t\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star \neq w_i \\ j\in U_i}} (g^{(a^{n^\star - y_t + j})b^2cd_k^5/d_t})^{\frac{1}{\sigma_k^\star - w_i}} \Bigg).
\end{aligned} \tag{51}$$

For $i = 1$, $g^{c_{6,1}}$ is computed exactly as above except only without the fourth line. For $i = \ell$, $g^{c_{6,\ell}}$ is computed exactly as above except only without the sixth line. By inspection, we can see that all the terms appearing in the computation are available from $\boldsymbol{D}$.

The equation is quite overwhelming at the first glance. However, they are written systematically as follows. The terms in the second to fourth line correspond to $g^{s_{i-1}(h_1+h_2 w_i^\star)}$ and those in the fifth to seventh line correspond to $g^{s_i(h_3+h_4 w_i^\star)}$ (after the cancellation across these two terms). Moreover, the second (resp., third, fourth) line corresponds to the first (resp., second, third) term of the programming of s_{i-1} in Eq. (42). Similarly, the fifth (resp., sixth, seventh) line corresponds to the first (resp., second, third) term of the programming of s_i in Eq. (42). We note that the programming of s_0 of Eq. (41) (resp., s_ℓ of Eq. (43)) can be viewed as that of s_i in Eq. (42) but without the third (resp., second) term; therefore, the computation of $g^{c_{6,0}}$ (resp., $g^{c_{6,\ell}}$) does not contain the fourth (resp., sixth) line.

(Re-randomizing Ciphertext). The simulated ciphertext is not perfectly distributed yet since the randomness $s, s_0, \ldots, s_\ell$ are still correlated. We will re-randomize $s_0, \ldots, s_\ell$, so that they are independent from s. This can be done by randomly choosing $s_0', \ldots, s_\ell' \xleftarrow{\$} \mathbb{Z}_p$ and computing a new ciphertext as

$$\begin{aligned}
g^{\tilde{s}} &= g^{s}, \\
g^{\tilde{s}\eta} &= g^{s\eta}, \\
g^{-\tilde{s}\phi_1+\tilde{s}_\ell\phi_2} &= g^{-s\phi_1+s_\ell\phi_2}(g^{\phi_2})^{s'_\ell}, \\
\forall_{i\in[0,\ell]} \qquad g^{\tilde{s}_i} &= g^{s_i}g^{s'_i}, \\
g^{\tilde{s}_0 h_0} &= g^{s_0 h_0}(g^{h_0})^{s'_0}, \\
\forall_{i\in[1,\ell]} \quad g^{\tilde{s}_{i-1}(h_1+h_2w^\star_i)+\tilde{s}_i(h_3+h_4w^\star_i)} &= g^{s_{i-1}(h_1+h_2w^\star_i)+s_i(h_3+h_4w^\star_i)}((g^{h_1})(g^{h_2})^{w^\star_i})^{s'_{i-1}}((g^{h_3})(g^{h_4})^{w^\star_i})^{s'_i}
\end{aligned}$$

where the new ciphertext has new randomness $\tilde{s}_i = s_i + s'_i$.

Guess. $\mathcal{A}$ will eventually output a guess if $\alpha = 0$ or random, our simulator $\mathcal{B}$ just outputs the guess that $\tau = 0$ and τ is random respectively. Hence, the advantage of $\mathcal{B}$ winning the $(n^\star, m^\star)$-EDHE2 game is exactly the same as the advantage of $\mathcal{A}$ winning the co-selective master-key hiding game. □

C Proof for Key-Policy over Doubly Spatial Encryption

C.1 Proof of Selective Security

In this section, we prove the selective master-key hiding security of Scheme 6, our pair encoding for KP-DSE (Theorem 11 in §7). Before the proof, we state some useful propositions.

Proposition 39. *Let $P = \begin{pmatrix} 1 & \mathbf{0} \\ \boldsymbol{c}^\top & P' \end{pmatrix} \in \mathsf{AffM}(\mathbb{Z}_p^{n\times d})$ be an affine matrix. Suppose that $(1, \boldsymbol{v}) \notin \mathsf{AffSp}(P)$. Then there is an efficient algorithm that outputs $\boldsymbol{w} \in \mathbb{Z}_p^n$ such that $\boldsymbol{w}(\boldsymbol{v} - \boldsymbol{c})^\top \neq 0$ and $\boldsymbol{w}P' = \mathbf{0}_d$.*

Let $M \in \mathbb{Z}_N^{d\times n}$, we recall the notation of a row space as $\mathsf{RowSp}(M) = \{\boldsymbol{w}M \mid \boldsymbol{w} \in \mathbb{Z}_N^d\}$. We also denote its null space as $\mathsf{NullSp}(M) = \{\boldsymbol{v} \in \mathbb{Z}_N^n \mid M\boldsymbol{v}^\top = 0\}$.

Proposition 40. *For a matrix $G \in \mathbb{Z}_N^{d\times n}$ with linearly independent rows, there exists an efficient algorithm that outputs $H \in \mathbb{Z}_N^{n\times(n-d)}$ such that $\mathsf{NullSp}(H^\top) = \mathsf{RowSp}(G)$. That is, $\boldsymbol{v} \in \mathsf{RowSp}(G)$ if and only if $\boldsymbol{v}H = \mathbf{0}_{n-d}$. We denote $H = G^{(\perp)}$. Moreover, we also have that $GG^{(\perp)} = 0^{d\times(n-d)}$.*[9]

Proposition 41. *Suppose that $\boldsymbol{a} = (a, a^2, \ldots, a^{n+1-d}), \boldsymbol{\sigma} = (a^n, a^{n-1}, \ldots, a^d) \in \mathbb{Z}_p^{n+1-d}$. Then for any $\boldsymbol{v}, \boldsymbol{x} \in \mathbb{Z}_p^{n+1-d}$, the coefficient of the term a^{n+1} contained in $(\boldsymbol{v}\sigma^\top)(\boldsymbol{a}\boldsymbol{x}^\top)$ is exactly $\boldsymbol{v}\boldsymbol{x}^\top$.*

We now describe the proof of Theorem 11.

Proof. Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the selective master-key hiding game against our pair encoding Scheme 6. We construct a simulator $\mathcal{B}$ that solves the (n, t)-EDHE3 assumption by running $\mathcal{A}$ and simulating the security game as follows.

Ciphertext Query. The selective game begins with $\mathcal{A}$ making a ciphertext query for $\Omega^\star = (Y^{\star(1)}, \ldots, Y^{\star(t)})$ where $Y^{\star(i)} \in \mathsf{AffM}(\mathbb{Z}_p^{n\times d_i})$ and $A \in \mathbb{Z}_p^{m\times k}$. $\mathcal{B}$ takes the (n, t)-EDHE3 challenge $(\boldsymbol{D}, T)$. Its task is to guess if $T = g^{a^{n+1}z}$ or T is a random element in $\mathbb{G}_p$.

[9] The pair $(G, G^{(\perp)})$ holds exactly the relation of "generator matrix" and "parity check matrix" in coding theory. There is an efficient conversion between each other (*e.g.*, [30](p.44)).

(Programming Parameters). $\mathcal{B}$ first compute for $i \in [1,t]$ an affine matrix $K_i \in \mathsf{AffM}(\mathbb{Z}_p^{(n-d_i)\times n})$ such that

$$K_i Y^{\star(i)} = \begin{pmatrix} 1 & \mathbf{0}_{d_i} \\ \mathbf{0}_{n-d_i}^{\top} & \mathbf{0} \end{pmatrix} \in \mathsf{AffM}(\mathbb{Z}_p^{(n-d_i)\times d_i}),$$

where here $\mathbf{0} \in \mathbb{Z}_p^{(n-d_i)\times d_i}$. This can be done as follows. Let $Y^{\star(i)} = \left(\begin{smallmatrix} 1 & \mathbf{0} \\ \boldsymbol{y}_i^{\top} & Y_i' \end{smallmatrix}\right)$. Then, we find a matrix K_i' such that $\mathsf{RowSp}(Y_i'^{\top}) = \mathsf{NullSp}(K_i')$ (see Proposition 40). We then set $K_i = \left(\begin{smallmatrix} 1 & \mathbf{0} \\ -K_i'\boldsymbol{y}_i^{\top} & K_i' \end{smallmatrix}\right)$. Intuitively, we project the affine space from dimension n to dimension $n - d_i$ in such a way that all points (vectors) in $\mathsf{AffSp}(Y^{\star(i)})$ maps to $(1, \mathbf{0}_{n-d_i})$ and all the other points outside it would not map to $(1, \mathbf{0}_{n-d_i})$.

Denote $\boldsymbol{a}^{(i)} = (a, a^2, \ldots, a^{n+1-d_i})$ and $\boldsymbol{\sigma}^{(i)} = (a^n, a^{n-1}, \ldots, a^{d_i})$. $\mathcal{B}$ computes the parameter by choosing $\bar{\boldsymbol{h}}' \xleftarrow{\$} \mathbb{Z}_p^{n+1}, \phi_1', \phi_2', \phi_3' \xleftarrow{\$} \mathbb{Z}_p$ then implicitly sets

$$\bar{\boldsymbol{h}} = \sum_{i\in[1,t]} (0, \frac{\boldsymbol{a}^{(i)}}{b_i^2})K_i + (\frac{a^n c}{b_i}, \mathbf{0}) + \bar{\boldsymbol{h}}', \quad \phi_1 = \phi_1' + a^n z \quad \phi_2 = \phi_2' - a^n, \quad \phi_3 = \phi_3' - a^n, \quad \eta = z,$$

Note that $g^{\bar{\boldsymbol{h}}}, g^{\phi_2}, g^{\phi_3}$ can be computed from $\boldsymbol{D}$ but $g^{a^n z}, g^z$ are unavailable.

(Programming Randomness in Ciphertext). We now describe how $\mathcal{B}$ produce a ciphertext for $\Omega^{\star}$. $\mathcal{B}$ first implicitly sets

$$s = c/z, \qquad w = c, \qquad \forall_{i\in[1,t]}\ s_i = b_i$$

(Cancellation in Ciphertext). From the above (implicit) definitions, the ciphertext is well defined. It can be computed due to the cancelation of unknown elements as follows.

- Canceling $g^{\boldsymbol{a}^{(i)}/b_i}$ in $g^{(w\phi_3,\mathbf{0})+s_i\bar{\boldsymbol{h}}Y^{\star(i)}}$ since $(b_i)((0, \frac{\boldsymbol{a}^{(i)}}{b_i^2})K_i)Y^{\star(i)} = (0, \frac{\boldsymbol{a}^{(i)}}{b_i}) \begin{pmatrix} 1 & \mathbf{0}_{d_i} \\ \mathbf{0}_{n-d_i}^{\top} & 0 \end{pmatrix} = \mathbf{0}.$
- Canceling $g^{a^n c}$ in $g^{(w\phi_3,\mathbf{0})+s_i\bar{\boldsymbol{h}}Y^{\star(i)}}$ by

$$g^{(c)(-a^n)} \text{ from } g^{(w)(\phi_3)}, \qquad g^{(b_i)(a^n c/b_i)} \text{ from } g^{(s_i)(\bar{\boldsymbol{h}})Y^{\star(i)}}$$

- Canceling $g^{a^n c}$ in $g^{s\phi_1+w\phi_2}$ by

$$g^{(c/z)(a^n z)} \text{ from } g^{(s)(\phi_1)}, \qquad g^{(b_i)(-a^n c)} \text{ from } g^{(w)(\phi_2)}$$

(Simulating Ciphertext). The ciphertext is then computable from remaining terms which consist of only known elements from $\boldsymbol{D}$ as

$$g^s = g^{c/z}, \quad g^{s\eta} = g^c, \quad g^w = g^c, \quad g^{s\phi_1+w\phi_2} = (g^s)^{\phi_1'}(g^w)^{\phi_2'}, \quad \forall_{i\in[1,t]}\ g^{s_i} = g^{b_i},$$

$$g^{(w\phi_3,\mathbf{0})+s_i\bar{\boldsymbol{h}}Y^{\star(i)}} = g^{(w\phi_3',\mathbf{0})}(g^{s_i})^{\bar{\boldsymbol{h}}'Y^{\star(i)}} \prod_{\substack{j\in[1,t] \\ \text{s.t. } j\neq i}} g^{(0,\frac{\boldsymbol{a}^{(j)}b_i}{b_j^2})K_j Y^{\star(i)}} g^{\frac{a^n c b_i}{b_j}}.$$

(Re-randomizing Ciphertext). The simulated ciphertext is not perfectly distributed yet since their randomness are still correlated. $\mathcal{B}$ re-randomizes the ciphertext so that all the other randomness are independent from s which we will not re-randomize since $\mathcal{B}$ does not possess g^{ϕ_1}, g^{η}. $\mathcal{B}$ does this by using the known $g^{\bar{\boldsymbol{h}}}, g^{\phi_2}, g^{\phi_3}$.

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

(Programming and Simulating Key). $\mathcal{B}$ will implicitly set $\alpha = \tau$, where we recall $T = g^{a^{n+1}z+\tau}$ from the problem instance. $\mathcal{B}$ also sets $r = a, u = 0$. This can be done by setting $g^{\alpha+\phi_1 r+u\eta} = T(g^r)^{\phi_1'}$. $\mathcal{B}$ sets $\boldsymbol{v} = a^{n+1}\boldsymbol{\nu} + (\phi_2' a, \mathbf{0})$. This is consistent with $v_1 = \phi_2 r = (-a^n + \phi_2')a$. $\mathcal{B}$ then constructs key components regarding $X^{(i)}$ for $i \in [1,m]$ as follows.

Case 1: $i \notin S$ (That is, for all $j \in [1,t]$, $R^{\mathsf{DS}}(X^{(i)}, Y^{\star(j)}) = 0$).

In this case, we have that for all $j \in [1,t]$, $\mathsf{AffSp}(X^{(i)}) \cap \mathsf{AffSp}(Y^{\star(j)}) = \emptyset$. Consider $j \in [1,t]$. Therefore, $\mathsf{AffSp}(K_jX^{(i)}) \cap \mathsf{AffSp}(K_jY^{\star(j)}) = \emptyset$, since both affine spaces after the projection by K_j would also not intersect. Hence $(1, \mathbf{0}_{n-d_j}) \notin \mathsf{AffSp}(K_jX^{(i)})$. We can write $K_jX^{(i)} = \begin{pmatrix} K_jX_1^{(i)} & K_jX_{[2,\ell_i]}^{(i)} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0} \\ (\boldsymbol{\kappa}^{(i,j)})^\top & J^{(i,j)} \end{pmatrix}$. Here, we denote $X_1^{(i)}$ as the first column vector of $X^{(i)}$, and $X_{[2,\ell_i]}^{(i)}$ is the sub-matrix of $X^{(i)}$ that excludes the first column.

Then, from Proposition 39, we can find $\boldsymbol{w}^{(i,j)} \in \mathbb{Z}_p^{n-d_j}$ such that $\boldsymbol{w}^{(i,j)}(\mathbf{0} - (\boldsymbol{\kappa}^{(i,j)}))^\top \neq 0$ and $\boldsymbol{w}^{(i,j)}J^{(i,j)} = \mathbf{0}_{\ell_i}$. We can scale $\boldsymbol{w}^{(i,j)}$ so that $\boldsymbol{w}^{(i,j)}(\boldsymbol{\kappa}^{(i,j)})^\top = -1$. $\mathcal{B}$ implicitly defines

$$r_i = A_i\boldsymbol{\nu}^\top \left(a + c \sum_{j\in[1,t]} b_j \boldsymbol{w}^{(i,j)}(\boldsymbol{\sigma}^{(j)})^\top \right).$$

From this assignment, the key component is well defined. It can be computed due to the cancelation of unknown elements as follows.

- Cancelation of $g^{a^{n+1}c/b_j}$ for all $j \in [1,t]$ in $g^{r_i\bar{\boldsymbol{h}}X^{(i)}}$ occurs as we can see that coefficient of $a^{n+1}c/b_j$ in the following exponents are $(A_i\boldsymbol{\nu}^\top, \mathbf{0})$ and $(-A_i\boldsymbol{\nu}^\top, \mathbf{0})$ respectively.

  $$g^{(A_i\boldsymbol{\nu}^\top a)((\frac{a^n c}{b_j}, \mathbf{0}))} \text{ from } g^{(r_i)(\bar{\boldsymbol{h}})X^{(i)}}, \quad g^{(A_i\boldsymbol{\nu}^\top cb_j\boldsymbol{w}^{(i,j)}(\boldsymbol{\sigma}^{(j)})^\top)((0, \frac{\boldsymbol{a}^{(j)}}{b_j^2})K_j)(X^{(i)})} \text{ from } g^{(r_i)(\bar{\boldsymbol{h}})(X^{(i)})}$$

  More precisely, we observe that

  $$\begin{aligned}(cb_j\boldsymbol{w}^{(i,j)}(\boldsymbol{\sigma}^{(j)})^\top)(0, \frac{\boldsymbol{a}^{(j)}}{b_j^2})K_jX^{(i)} &= (c\boldsymbol{w}^{(i,j)}(\boldsymbol{\sigma}^{(j)})^\top)(0, \frac{\boldsymbol{a}^{(j)}}{b_j}) \begin{pmatrix} 1 & \mathbf{0} \\ (\boldsymbol{\kappa}^{(i,j)})^\top & J^{(i,j)} \end{pmatrix} \\ &= \left(c\boldsymbol{w}^{(i,j)}(\boldsymbol{\sigma}^{(j)})^\top \frac{\boldsymbol{a}^{(j)}}{b_j}(\boldsymbol{\kappa}^{(i,j)})^\top,\ c\boldsymbol{w}^{(i,j)}(\boldsymbol{\sigma}^{(j)})^\top \frac{\boldsymbol{a}^{(j)}}{b_j} J^{(i,j)} \right),\end{aligned}$$

  and from Proposition 41, we have that the coefficient of $a^{n+1}c/b_j$ in this term is exactly $(-1, \mathbf{0})$.

- Canceling $g^{a^{n+1}}$ in $g^{A_i\boldsymbol{v}^\top + r_i\phi_3}$ by

  $$g^{(A_i)(-a^{n+1}\boldsymbol{\nu}^\top)} \text{ from } g^{(A_i)(\boldsymbol{v}^\top)}, \qquad g^{(-A_i\boldsymbol{\nu}^\top a)(a^n)} \text{ from } g^{(r_i)(\phi_3)}$$

The key is then computable from remaining terms which consist of only known elements from $\boldsymbol{D}$ as

$$g^{A_i\boldsymbol{\alpha}^\top + r_i\phi_3} = g^{A_i\boldsymbol{\nu}^\top \sum_{j\in[1,t]} cb_jP_j(a)}(g^{r_i})^{\phi'_3}, \qquad g^{r_i} = g^{A_i\boldsymbol{\nu}^\top(a+\sum_{j\in[1,t]} cb_jR_j(a))}$$

where $P_j(a), R_j(a)$ is a polynomial with degree at most $2n$. Due to the above cancelation, the element $g^{r_i\bar{\boldsymbol{h}}X^{(i)}}$ can be computed using the term of the forms $g^{a^k/b'^2_j}, g^{a^kc^2b_j/b'_j}$ and of the form $g^{a^kcb_j/b'^2_j}$ with $j \neq j'$ or $k \neq n+1$.

Case 2: $i \in S$ (That is, there exists $j \in [1,t]$ where $R^{\mathsf{DS}}(X^{(i)}, Y^{\star(j)}) = 1$).

In this case, we have $A_i\boldsymbol{v}^\top = a^{n+1}A_i\boldsymbol{\nu}^\top + A_{i,1}\phi'_2a = A_{i,1}\phi'_2a$. We simply create key components by choosing $r_i \xleftarrow{\$} \mathbb{Z}_p$ and simply computing $g^{r_i}, g^{r_i\bar{\boldsymbol{h}}X^{(i)}}$ and $g^{A_i\boldsymbol{v}^\top + r_i\phi_3} = (g^a)^{A_{i,1}\phi'_2}(g^{\phi_3})^{r_i}$.

(Re-randomizing Key). The simulated key is not perfectly distributed yet since their randomness are still correlated. $\mathcal{B}$ re-randomizes the key by choosing randomly r'_i, u'_i for $i \in [1,m]$ and $\tau'_2, \dots, \tau'_k \xleftarrow{\$} \mathbb{Z}_p$. Let $\boldsymbol{\tau}' = (0, \tau'_2, \dots, \tau'_k)$. It computes a new key as follows.

$$\begin{aligned}
g^{\alpha+\tilde{r}\phi_1+\tilde{u}\eta} &= g^{\alpha+r\phi_1+u\eta}(g^c)^{u'}(g^{r'})^{\phi'_1},\\
g^{\tilde{u}} &= g^u(g^{c/z})^{u'}(g^{a^n})^{r'},\\
g^{\tilde{r}} &= g^rg^{r'},\\
g^{A_i\tilde{\boldsymbol{v}}^\top+\tilde{r}_i\phi_3} &= g^{A_i\boldsymbol{v}^\top+r_i\phi_3}g^{A_i\boldsymbol{\tau}'^\top}(g^{\phi_3})^{r'_i}\\
g^{\tilde{r}_i} &= g^{r_i}g^{r'_i},\\
g^{\tilde{r}_i\bar{\boldsymbol{h}}X^{(i)}} &= g^{r_i\bar{\boldsymbol{h}}X^{(i)}}g^{r'_i\bar{\boldsymbol{h}}X^{(i)}}
\end{aligned}$$

where the new key has new randomness $\tilde{r} = r + r'$, $\tilde{u} = u + (c/z)u' + a^nr'$, $\tilde{r}_i = r_i + r'_i$, and $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \boldsymbol{\tau}'$.

Guess. $\mathcal{A}$ will eventually output a guess if $\alpha = 0$ or random, our simulator $\mathcal{B}$ just outputs the guess that $\tau = 0$ and τ is random respectively. Hence, the advantage of $\mathcal{B}$ winning the (n,t)-$\mathsf{EDHE3}$ assumption is exactly the same as the advantage of $\mathcal{A}$ winning the selective master-key hiding game. □

C.2 Proof of Co-Selective Security

In this section, we prove the co-selective master-key hiding security of Scheme 6, our pair encoding for KP-DSE (Theorem 12 in §7).

Proof. Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the co-selective master-key hiding game against our pair encoding Scheme 6. We construct a simulator $\mathcal{B}$ that solves the (n,m,k)-$\mathsf{EDHE4}$ assumption by running $\mathcal{A}$ and simulating the security game as follows.

Key Query. The co-selective game begins with $\mathcal{A}$ making a private key query for $\mathbb{A}^\star = (A^\star, \{X^{\star(i)}\}_{i\in[1,m]})$ where $X^{\star(i)} \in \mathsf{AffM}(\mathbb{Z}_p^{n\times d_i})$ and $A \in \mathbb{Z}_p^{m\times k}$. $\mathcal{B}$ takes the (n,m,k)-$\mathsf{EDHE4}$ challenge $(\boldsymbol{D}, T)$, where $T = g^{\tau+xcz}$. Its task is to guess if $\tau = 0$ or τ is random in $\mathbb{Z}_p$.

(Programming Parameters). $\mathcal{B}$ first compute for $i \in [1,m]$ an affine matrix $K_i \in \mathsf{AffM}(\mathbb{Z}_p^{(n-d_i)\times n})$ such that

$$K_iX^{\star(i)} = \begin{pmatrix} 1 & \boldsymbol{0}_{d_i} \\ \boldsymbol{0}^\top_{n-d_i} & \boldsymbol{0} \end{pmatrix} \in \mathsf{AffM}(\mathbb{Z}_p^{(n-d_i)\times d_i}),$$

where here $\mathbf{0} \in \mathbb{Z}_p^{(n-d_i)\times d_i}$. This can be done as previously described in the selective proof. Intuitively, we project the affine space from dimension n to dimension $n - d_i$ in such a way that all points (vectors) in $\mathsf{AffSp}(X^{\star(i)})$ maps to $(1, \mathbf{0}_{n-d_i})$ and all the other points outside it would not map to $(1, \mathbf{0}_{n-d_i})$.

We denote the following vectors to be used throughput the proof.

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

$\mathcal{B}$ computes the parameter by choosing $\bar{\boldsymbol{h}}' \xleftarrow{\$} \mathbb{Z}_p^{n+1}, \phi_1', \phi_2', \phi_3' \xleftarrow{\$} \mathbb{Z}_p$ then implicitly sets

$$\alpha = \tau, \qquad \phi_1 = zx + \phi_1', \qquad \phi_2 = a^{n+1}x + \phi_2', \qquad \eta = z$$
$$\bar{\boldsymbol{h}} = \sum_{i\in[1,m]} (A_i^\star \boldsymbol{x}^\top)(0, \frac{\boldsymbol{a}^{(i)}}{b_i^2})K_i + \bar{\boldsymbol{h}}', \qquad \phi_3 = \sum_{i\in[1,m]} -(A_i^\star \boldsymbol{x}^\top)\frac{a^{n+1}}{b_i} + \phi_3'.$$

We note that $g^{\bar{\boldsymbol{h}}}, g^{\phi_2}, g^{\phi_3}$ can be computed from $\boldsymbol{D}$ but $\mathcal{B}$ cannot compute g^{ϕ_1}, g^{η}.

(Programming Randomness in Key). We now describe how $\mathcal{B}$ produce a key for $\Omega^\star$. $\mathcal{B}$ first chooses $u' \xleftarrow{\$} \mathbb{Z}_p, \boldsymbol{v}' \xleftarrow{\$} \mathbb{Z}_p^k$ with $v_1' = 0$. It then implicitly sets

$$r = c, \qquad u = u'x^k a^{n+1}/z, \qquad \forall_{i\in[1,m]}\ r_i = cb_i, \qquad \boldsymbol{v} = a^{n+1}c\boldsymbol{x} + (\phi_2' c, \mathbf{0}) + \boldsymbol{v}',$$

where v_1 correctly obeys $v_1 = r\phi_2$.

(Cancellation in Key). From the above (implicit) definition, the key is well defined. It can be computed due to the cancelation of unknown elements as follows.

- Cancelation of $g^{x^j c\boldsymbol{a}^{(i)}/b_i}$ terms in $g^{r_i \bar{\boldsymbol{h}} X^{\star(i)}}$ occurs due to

$$(cb_i)((A_i^\star \boldsymbol{x}^\top)(0, \frac{\boldsymbol{a}^{(i)}}{b_i^2})K_i)X^{\star(i)} = c(A_i^\star \boldsymbol{x}^\top)(0, \frac{\boldsymbol{a}^{(i)}}{b_i})\begin{pmatrix} 1 & \mathbf{0}_{d_i} \\ \mathbf{0}_{n-d_i}^\top & 0 \end{pmatrix} = \mathbf{0}.$$

- Canceling $g^{cx^j a^{n+1}}$ in $g^{A_i \boldsymbol{v}^\top + r_i\phi_3 + u\eta}$ by

$$g^{(A_i)(a^{n+1}c\boldsymbol{x}^\top)} \text{ from } g^{(A_i)(\boldsymbol{v}^\top)}, \qquad g^{(cb_i)(-A_i\boldsymbol{x}^\top a^{n+1}/b_i)} \text{ from } g^{(r_i)(\phi_3)}$$

(Simulating Key). The key is then computable from remaining terms which consist of only known elements from $\boldsymbol{D}$ as

$$g^{\alpha + r\phi_1 + u\eta} = T(g^r)^{\phi_1'}(g^{x^k a^{n+1}})^{u'}$$
$$g^r = g^c,$$
$$g^u = (g^{x^k a^{n+1}/z})^{u'},$$
$$\forall_{i\in[1,m]}\ g^{A_i\boldsymbol{v}^\top + r_i\phi_3} = g^{A_i\boldsymbol{v}'^\top}(g^{r_i})^{\phi_3'} \prod_{\substack{j\in[1,m] \\ \text{s.t. } j\neq i}} g^{-(A_i^\star \boldsymbol{x}^\top)\frac{a^{n+1}b_i c}{b_j}}$$
$$\forall_{i\in[1,m]}\ g^{r_i} = g^{cb_i},$$
$$\forall_{i\in[1,m]}\ g^{r_i\bar{\boldsymbol{h}}X^{\star(i)}} = (g^{r_i})^{\bar{\boldsymbol{h}}'X^{\star(i)}} \prod_{\substack{j\in[1,m] \\ \text{s.t. } j\neq i}} g^{(A_i^\star \boldsymbol{x}^\top)(0, \frac{\boldsymbol{a}^{(j)}b_i c}{b_j^2})K_j X^{\star(i)}}.$$

The key is properly distributed due to the randomness $c, b_i, u', \phi_2', \boldsymbol{v}'$.

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

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

$$s = x^k a^{n+1}/z, \qquad\qquad w = \boldsymbol{\chi}\boldsymbol{\nu}^\top.$$

Then, for $j \in [1, t]$, $\mathcal{B}$ constructs the corresponding ciphertext element as follows. Consider $i \notin S$. We have that $R^{\mathsf{DS}}(X^{\star(i)}, Y^{(j)}) = 0$. That is, $\mathsf{AffSp}(X^{\star(i)}) \cap \mathsf{AffSp}(Y^{(j)}) = \emptyset$. Hence $\mathsf{AffSp}(K_i X^{\star(i)}) \cap \mathsf{AffSp}(K_i Y^{(j)}) = \emptyset$, since both affine spaces after the projection by K_i would also not intersect. Therefore $(1, \mathbf{0}_{n-d_i}) \notin \mathsf{AffSp}(K_i Y^{(j)})$. We can write $K_i Y^{(j)} = \begin{pmatrix} K_i Y_1^{(j)} & K_i Y_{[2,\ell_j]}^{(j)} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0} \\ (\boldsymbol{\kappa}^{(j,i)})^\top & J^{(j,i)} \end{pmatrix}$. Then, from Proposition 39, we can find $\boldsymbol{w}^{(j,i)} \in \mathbb{Z}_p^{n-d_i}$ such that $\boldsymbol{w}^{(j,i)}(\mathbf{0} - (\boldsymbol{\kappa}^{(j,i)}))^\top \neq 0$ and $\boldsymbol{w}^{(j,i)} J^{(j,i)} = \mathbf{0}_{\ell_j}$. We can scale $\boldsymbol{w}^{(j,i)}$ so that $\boldsymbol{w}^{(j,i)}(\boldsymbol{\kappa}^{(j,i)})^\top = 1$. $\mathcal{B}$ then implicitly sets

$$\forall_{j \in [1,t]}\ s_j = (\boldsymbol{\chi}\boldsymbol{\nu}^\top)\Big(\prod_{\substack{i \in [1,m] \\ \text{s.t. } i \notin S}} b_i \boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top\Big).$$

(Cancellation and Simulation for Ciphertext). From these (implicit) definitions, the ciphertext is well defined. It can be computed due to the cancelation of unknown elements as follows.

- Cancelation of $g^{a^{n+1}x^{k+1}/b_i}$ for all $i \in [1, m]$ in $g^{(w\phi_3, \mathbf{0}) + s_j \bar{\boldsymbol{h}} Y^{(j)}}$ can be categorized into two types:
  - Case $i \in S$. In this case, we claim that the coefficient of $a^{n+1}x^{k+1}/b_i$ in the exponent of the following term is exactly $A_i^\star \boldsymbol{\nu}^\top$, but then this is 0 since $i \in S$, hence the term vanishes:

    $$g^{(\boldsymbol{\chi}\boldsymbol{\nu}^\top)(-(A_i^\star \boldsymbol{x}^\top)\frac{a^{n+1}}{b_i})} \text{ from } g^{(w)(\phi_3)}.$$

    The claim then follows by applying Proposition 41 to the definition of $\boldsymbol{x}$ and $\boldsymbol{\chi}$ and seeing that the coefficient of x^{k+1} is $A_i^\star \boldsymbol{\nu}^\top a^{n+1}$. We note that since $i \in S$, the term $a^{n+1}x^{k+1}/b_i$ does not appear in $s_j \bar{\boldsymbol{h}} Y^{(j)}$ since in s_j the range in the product excludes the case where $i \in S$.
  - Case $i \notin S$. In this case, we claim that the coefficient of $a^{n+1}x^{k+1}/b_i$ in the following exponents are $-A_i \boldsymbol{\nu}^\top$ and $(A_i \boldsymbol{\nu}^\top, \mathbf{0})$ respectively:

    $$g^{(\boldsymbol{\chi}\boldsymbol{\nu}^\top)(-(A_i^\star \boldsymbol{x}^\top)\frac{a^{n+1}}{b_i})} \text{ from } g^{(w)(\phi_3)}, \quad g^{((\boldsymbol{\chi}\boldsymbol{\nu}^\top) b_i \boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top ((A_i^\star \boldsymbol{x}^\top)(0, \frac{\boldsymbol{a}^{(i)}}{b_i^2}) K_i)(Y^{(j)})} \text{ from } g^{(s_j)(\bar{\boldsymbol{h}})(Y^{(j)})}.$$

    We already prove the claim for the the former term. To prove the claim for the latter term, we will use two layers of applications of Proposition 41. The first applies to the pair of $\boldsymbol{x}$ and $\boldsymbol{\chi}$, where we can deduce that the coefficient of x^{k+1} is

    $$\begin{aligned} (A_i^\star \boldsymbol{\nu}^\top) b_i \boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top (0, \frac{\boldsymbol{a}^{(i)}}{b_i^2}) K_i Y^{(i)} &= (A_i^\star \boldsymbol{\nu}^\top) b_i \boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top (0, \frac{\boldsymbol{a}^{(i)}}{b_i^2}) \begin{pmatrix} 1 & \mathbf{0} \\ (\boldsymbol{\kappa}^{(j,i)})^\top & J^{(j,i)} \end{pmatrix} \\ &= (A_i^\star \boldsymbol{\nu}^\top) \left(\boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top \frac{\boldsymbol{a}^{(i)}}{b_i} (\boldsymbol{\kappa}^{(j,i)})^\top,\ \boldsymbol{\sigma}^{(i)} (\boldsymbol{w}^{(j,i)})^\top \frac{\boldsymbol{a}^{(i)}}{b_i} J^{(j,i)} \right). \end{aligned}$$

We then apply Proposition 41 again but now to the pair of $\boldsymbol{a}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$, where we can deduce that the coefficient of a^{n+1} is

$$(A_i^\star \boldsymbol{\nu}^\top)\left((\boldsymbol{w}^{(j,i)})^\top (\boldsymbol{\kappa}^{(j,i)})^\top, (\boldsymbol{w}^{(j,i)})^\top J^{(j,i)}\right) = (A_i^\star \boldsymbol{\nu}^\top)(1, \mathbf{0}).$$

- Canceling $g^{x^{k+1}a^{n+1}}$ in $g^{s\phi_1 + w\phi_2}$ by

  $$g^{(-\nu_1 x^k a^{n+1}/z)(x)} \text{ from } g^{(s)(\phi_1)}, \qquad\qquad g^{(x^k \nu_1)(x a^{n+1})} \text{ from } g^{(w)(\phi_2)},$$

  where we recall that indeed $\nu_1 = -1$.

Due to the cancelations above, the ciphertext can be computed from available terms from $\boldsymbol{D}$.

(Re-randomizing Ciphertext). The simulated ciphertext is not perfectly distributed yet since their randomness are still correlated. $\mathcal{B}$ re-randomizes the ciphertext so that all the other randomness are independent from s which we will not re-randomize since $\mathcal{B}$ does not possess g^z. $\mathcal{B}$ chooses randomly s'_j for $j \in [1, t]$ and $w' \xleftarrow{\$} \mathbb{Z}_p$. It can compute a new ciphertext elements by using the known $g^{\bar{\boldsymbol{h}}}, g^{\phi_3}, g^{\phi_2}$.

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

D Proof for Dual FE for Regular Languages

In this section, we prove the security of Scheme 7, our pair encoding for dual FE for regular languages, described in §8.2.

D.1 Proof of Selective Security

The selective master-key hiding security of Scheme 7 is proved under the new assumption which is similar to EDHE2 (Definition 4). We recall that EDHE2 was used to prove the *co-selective* master-key hiding security of Scheme 3, our encoding for FE for regular languages. We call the new assumption EDHE2-Dual. It is defined exactly as EDHE2 except only two differences: a given element $g^{a^{n-1}c/z}$ in the EDHE2 assumption is substituted with $g^{b/z}$, and the target element g^{abz} is substituted with $g^{a^n cz}$. These two differences are emphasized in the box in the following precise definition for clarity. The generic security of the assumption can be shown similarly (see Lemma 46).

Definition 9 ((n, m)-EDHE2-Dual Assumption)**.** The (n, m)-Expanded Diffie-Hellman Exponent Assumption-2-Dual in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, b, c, d_1, \ldots, d_m, z \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$ (for simplicity). Suppose that an adversary is given a target element $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of

$$\begin{aligned}
& g, g^a, g^b, \boxed{g^{b/z}} \\
\forall_{i\in[1,n],\ j,j'\in[1,m],j\neq j'} \quad & g^{a^i/d_j^2}, g^{a^i b/d_j}, g^{d_j}, g^{a^i d_j/d_{j'}^2}, g^{a^i b d_j/d_{j'}}, g^{a^i/d_j^6}, g^{a^i d_j/d_{j'}^6}, \\
\forall_{i\in[0,n-1]} \quad & g^{a^i c}, g^{a^i b c d_j}, \\
\forall_{i\in[0,n],\ j\in[1,m]} \quad & g^{a^i b c d_j^5}, \\
\forall_{i\in[1,2n-1],\ j,j'\in[1,m],j\neq j'} \quad & g^{a^i b c d_j/d_{j'}^2}, g^{a^i b c d_j^5/d_{j'}^6}, \\
\forall_{i\in[1,2n-1],i\neq n\ j\in[1,m]} \quad & g^{a^i bc/d_j}, \\
\forall_{i\in[1,2n-1],\ j,j'\in[1,m]} \quad & g^{a^i c/d_j^2}, g^{a^i b^2 c d_j/d_{j'}}, g^{a^i b c d_j/d_{j'}^6}, g^{a^i c/d_j^6}, g^{a^i b c d_j^5/d_{j'}^2}, g^{a^i b^2 c d_j^5/d_{j'}}
\end{aligned}$$

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

The proof of Theorem 19 follows almost exactly from the proof of Theorem 8 for the co-selective security of Scheme 3 under the EDHE2. For compactness and clarity of the presentation, we point out exactly where in the simulation is the same and refer the reader to the proof of Theorem 8.

Proof (of Theorem 19). Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the selective master-key hiding game against our pair encoding Scheme 7 for the dual DFA-based predicate. We construct a simulator $\mathcal{B}$ that solves the (n,m)-EDHE2-Dual by running $\mathcal{A}$ and simulating the security game as follows.

Ciphertext Query. The selective game begins with $\mathcal{A}$ making a ciphertext query for a DFA $M^\star = (Q^\star, \mathcal{T}^\star, q_0, q_{n^\star-1})$. We parse $\mathcal{T}^\star = \{(x_1, y_1, \sigma_1^\star), \ldots, (x_{m^\star}, y_{m^\star}, \sigma_{m^\star}^\star)\}$, where we denote $m^\star = |\mathcal{T}^\star|$. $\mathcal{B}$ takes as an input the $(n^\star, m^\star)$-EDHE2-Dual challenge $(\boldsymbol{D}, T)$. Its task is to guess whether $T = g^{a^n cz}$ or T is a random element in $\mathbb{G}_p$. That is, if we denote $T = g^{\tau + a^n cz}$, the task is to guess $\tau = 0$ or τ is a random element in $\mathbb{Z}_p$. $\mathcal{B}$ will implicitly define $\alpha = \tau$ by embedding T in the simulation for keys (see below).

(Programming Parameters). $\mathcal{B}$ computes the parameters $g^{h_0}, g^{h_1}, \ldots, g^{h_4}, g^{\phi_2}$ explicitly in exactly the same manner as in Eq. (30). $\mathcal{B}$ chooses $\phi_1', \phi' \xleftarrow{\$} \mathbb{Z}_p$ then sets the parameters

$$g^{\phi_1} = g^a g^{\phi_1'},$$

and then *implicitly* defines

$$g^{\phi} = g^{az} g^{\phi'} \qquad\qquad g^{\eta} = g^{z}$$

We note that the programming of η is the same as in Eq. (31), while that of ϕ_1 is different. We also note that ϕ does not appear in the proof of Theorem 8, by construction.

(Programming Randomness in Ciphertext). $\mathcal{B}$ first implicitly sets $s = -b/z$. $\mathcal{B}$ then implicitly sets u, s_0, v, s_t for $t \in [1,m]$, u_x for $q_x \in Q^\star \setminus \{q_{n^\star-1}\}$ as in Eq. (32), where we replace the variables r_0, r, r_t, u_x there with s_0, v, s_t, u_x here respectively. In other words, we sets

$$s_0 = d_1, \qquad v = b, \qquad \forall_{t\in[1,m^\star]}\ \ s_t = d_t, \qquad \forall_{q_x\in Q^\star\setminus\{q_{n^\star-1}\}}\ \ u_x = (a^{n^\star-x})b.$$

Now recall that $u_{n^\star-1} = \phi_2 v$, hence from our programming we have $u_{n^\star-1} = ab$. Therefore, the above equation for u_x indeed holds for all $q_x \in Q^\star$.

(Cancellation in Ciphertext). The cancellation in ciphertext here is exactly the same as the cancellation in key in the proof of Theorem 8 (where we replace the notation for variables properly), with one additional cancellation as follows.

- Canceling g^{ab} in $g^{s\phi+v\phi_1}$ by

$$g^{(-b/z)(az)} \text{ from } g^{(s)(\phi)}, \qquad\qquad g^{(b)(a)} \text{ from } g^{(v)(\phi_1)}$$

(Simulating Ciphertext). The simulation for ciphertext here is exactly the same as the simulation for the corresponding terms in key in the proof of Theorem 8. That is, we simulate as in Eq. (35)- (40), where we replace the notation for variables properly. The additional terms are simulated as follows.

$$\begin{aligned} g^{s\phi+v\phi_1} &= (g^{b/z})^{-\phi'}(g^b)^{\phi'_1} \\ g^s &= (g^{b/z})^{-1} \\ g^{s\eta} &= (g^b)^{-1}. \end{aligned}$$

(Re-randomizing Ciphertext). We can re-randomize $(v, s_0, \{s_t\}_{t\in[1,m]}, \{u_x\}_{q_x\in Q^\star})$ since $\mathcal{B}$ knows $g^{\phi_1}, g^{h_0}, \ldots, g^{h_4}$. This makes these randomness independent from s.

Key Query. The adversary $\mathcal{A}$ makes a key query for string $\boldsymbol{w} = (w_1, \ldots, w_\ell)$. $\mathcal{B}$ creates the key for $\boldsymbol{w}$ as follows. We use the same notation as in the proof of Theorem 8 as follows. We denote by $\boldsymbol{w}_i$ the vector formed by the last $\ell - i$ symbol of $\boldsymbol{w}$. That is $\boldsymbol{w}_i = (w_{i+1}, \ldots, w_\ell)$. Hence $\boldsymbol{w}_0 = \boldsymbol{w}$ and $\boldsymbol{w}_\ell$ is the empty string. For $q_k \in \{q_0, \ldots, q_{n^\star-1}\} = Q$, let $M_k^\star$ be the same DFA as $M^\star$ except that the start state is set to q_k. Then, for each $i \in [0, \ell]$ we define $U_i = \{k \in [0, n^\star - 1] \mid R(M_k^\star, \boldsymbol{w}_i) = 1\}$. From this and the query restriction that $R(M^\star, \boldsymbol{w}) = 0$, we have $0 \notin U_0$. Due to the WLOG condition, we have $U_\ell = \{n^\star - 1\}$.

(Programming Randomness in Key). $\mathcal{B}$ implicitly defines r_0, r_i for $i \in [1, \ell-1]$, and r_ℓ exactly as in Eq. (41), (42), and (43), respectively, in the proof of Theorem 8, where we replace the variables s_i there with r_i here respectively for $i \in [0, \ell]$. $\mathcal{B}$ also implicitly defines $u = 0$ and

$$r = a^{n^\star-1}c.$$

We note that only this term differs from the programming of s in the proof of Theorem 8.

(Cancellation in Key). The cancellation in key here is exactly the same as the cancellation in ciphertext in the proof of Theorem 8 (where we replace the notation for variables properly) albeit with one difference as follows.

- Canceling $g^{a^{n^\star}c}$ in $g^{-r\phi_1+r_\ell\phi_2}$ by

$$g^{(a^{n^\star-1}c)(a)} \text{ from } g^{(r)(\phi_1)}, \qquad\qquad g^{(a^{n^\star-1}c)(a)} \text{ from } g^{(r_\ell)(\phi_2)}$$

(Simulating Key). The simulation for key here is exactly the same as the simulation for the corresponding terms in ciphertext in the proof of Theorem 8. That is, we simulate as in Eq. (46)-

(51), where we replace the notation for variables properly. The other terms are simulated as follows.

$$\begin{aligned} g^{r} &= g^{a^{n^\star-1}c}, \\ g^{\alpha+r\phi+u\eta} &= T(g^{a^{n^\star-1}c})^{\phi'} \\ g^{-r\phi_1+r_\ell\phi_2} &= (g^{a^{n^\star-1}c})^{-\phi_1'} \cdot \prod_{\substack{k\in[1,m^\star] \\ \text{s.t. } \sigma_k^\star\neq w_\ell}} (g^{a^{n^\star}bcd_k^5})^{\frac{1}{\sigma_k^\star-w_\ell}}, \\ g^{u} &= 1. \end{aligned}$$

We note that T is embedded in such a way that $\tau = \alpha$.

(Re-randomizing Key). We can re-randomize r_i for $i \in [0, \ell]$ since $\mathcal{B}$ knows $g^{\phi_1}, g^{h_0}, \ldots, g^{h_4}$. A non-trivial term to re-randomize is r since $\mathcal{B}$ does not know g^{ϕ}. We re-randomize this by using u to cancel out unknown terms. That is, $\mathcal{B}$ chooses $r', u' \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets the new randomness to $\tilde{r} = r + r'$ and $\tilde{u} = u - ar' + (b/z)u'$.

Guess. $\mathcal{A}$ will eventually output a guess if $\alpha = 0$ or random, our simulator $\mathcal{B}$ just outputs the guess that $\tau = 0$ and τ is random respectively. Hence, the advantage of $\mathcal{B}$ winning the $(n^\star, m^\star)$-EDHE2-Dual game is exactly the same as the advantage of $\mathcal{A}$ winning the selective master-key hiding game. □

D.2 Proof of Co-Selective Security

Analogously to the previous section, the co-selective master-key hiding security of Scheme 7 is proved under the new assumption which is similar to EDHE1 (Definition 3). We recall that EDHE1 was used to prove the *selective* master-key hiding security of Scheme 3, our encoding for FE for regular languages. We call the new assumption EDHE1-Dual. It is defined exactly as EDHE1 except only two differences: a given element $g^{a^\ell c/z}$ in the EDHE1 assumption is substituted with $g^{b/z}$, and the target element g^{abz} is substituted with $g^{a^{\ell+1}cz}$. These two differences are emphasized in the box in the following precise definition for clarity. The generic security of the assumption can be shown similarly (see Lemma 44).

Definition 10 (ℓ-EDHE1-Dual Assumption)**.** The ℓ-Expanded Diffie-Hellman Exponent Assumption-1-Dual in subgroup is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$. Let $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Let $a, b, c, d_1, \ldots, d_{\ell+1}, f, z \xleftarrow{\$} \mathbb{Z}_N$. Denote $g = g_2$ and $p = p_2$ (for simplicity). Suppose that an adversary is given a target element $T \in \mathbb{G}_p$, and $\boldsymbol{D}$ consisting of

$$\begin{aligned} & g, g^{a}, g^{b}, g^{a/f}, g^{1/f}, \boxed{g^{b/z}} \\ \forall_{i\in[1,\ell+1]} \quad & g^{a^i/d_i}, g^{a^i bf} \\ \forall_{i\in[0,\ell]} \quad & g^{a^i c}, g^{bd_i}, g^{bd_i/f}, g^{abd_i/f} \\ \forall_{i\in[1,2\ell+1], i\neq\ell+1,\ j\in[1,\ell+1]} \quad & g^{a^i c/d_j} \\ \forall_{i\in[2,2\ell+2],\ j\in[1,\ell+1]} \quad & g^{a^i bf/d_j} \\ \forall_{i,j\in[1,\ell+1], i\neq j} \quad & g^{a^i bd_j/d_i} \end{aligned}$$

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

Proof (of Theorem 20). Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage in the co-selective master-key hiding game against our pair encoding Scheme 7 for the dual DFA-based predicate. We construct a simulator $\mathcal{B}$ that solves the ℓ-$\mathsf{EDHE1}$-Dual by running $\mathcal{A}$ and simulating the security game as follows.

Key Query. The co-selective game begins with $\mathcal{A}$ making a key query $\boldsymbol{w}^\star = (w_1^\star, \dots, w_{\ell^\star}^\star)$. $\mathcal{B}$ takes as an input the $\ell^\star$-$\mathsf{EDHE1}$-Dual challenge $(\boldsymbol{D}, T)$. Its task is to guess whether $T = g^{a^{\ell^\star+1}cz}$ or T is a random element in $\mathbb{G}_p$. That is, if we denote $T = g^{\tau + a^{\ell^\star+1}cz}$, the task is to guess $\tau = 0$ or τ is a random element in $\mathbb{Z}_p$. $\mathcal{B}$ will implicitly define $\alpha = \tau$ by embedding T in the simulation for keys (see below).

(Programming Parameters). $\mathcal{B}$ computes the parameters $g^{h_0}, g^{h_1}, \dots, g^{h_4}, g^{\phi_2}$ explicitly in exactly the same manner as in Eq. (9). $\mathcal{B}$ chooses $\phi_1', \phi' \xleftarrow{\$} \mathbb{Z}_p$ then sets the parameters

$$g^{\phi_1} = g^a g^{\phi_1'},$$

and then *implicitly* defines

$$g^{\phi} = g^{az} g^{\phi'} \qquad\qquad g^{\eta} = g^{z}$$

We note that the programming of η is the same as in Eq. (10), while that of ϕ_1 is different. We also note that ϕ does not appear in the proof of Theorem 7, by construction.

(Programming Randomness in Key). $\mathcal{B}$ implicitly sets $u = 0$ and

$$r = a^{\ell^\star} c, \qquad\qquad \forall_{i=[0,\ell^\star]} \;\; r_i = a^i c.$$

We note that the programming of r_i here is the same as s_i there in the proof of Theorem 7, while that of r here differs from that of s there.

(Cancellation in Key). The cancellation in key here is exactly the same as that in ciphertext in the proof of Theorem 7 (where we replace the notation for variables properly) albeit with one difference as follows.

- Canceling $g^{a^{\ell^\star+1}c}$ in $g^{-r\phi_1 + r_\ell \phi_2}$ by

$$g^{(a^{\ell^\star}c)(a)} \text{ from } g^{(r)(\phi_1)}, \qquad\qquad g^{(a^{\ell^\star}c)(a)} \text{ from } g^{(r_\ell)(\phi_2)}$$

(Simulating Key). The simulation for key here is exactly the same as the simulation for the corresponding terms in ciphertext in the proof of Theorem 7. That is, we simulate as in Eq. (13)-(16), where we replace the notation for variables properly. The other terms are simulated as follows.

$$\begin{aligned} g^{r} &= g^{a^{\ell^\star}c}, \\ g^{\alpha + r\phi + u\eta} &= T (g^{a^{\ell^\star}c})^{\phi'} \\ g^{-r\phi_1 + r_\ell\phi_2} &= (g^{a^{\ell^\star}c})^{-\phi_1'}, \\ g^{u} &= 1. \end{aligned}$$

We note that T is embedded in such a way that $\tau = \alpha$.

(Re-randomizing Key). We can re-randomize r_i for $i \in [0, \ell^\star]$ since $\mathcal{B}$ knows $g^{\phi_1}, g^{h_0}, \dots, g^{h_4}$. A non-trivial term to re-randomize is r since $\mathcal{B}$ does not know g^{ϕ}. We re-randomize this by using u to cancel out unknown terms. That is, $\mathcal{B}$ chooses $r', u' \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets the new randomness to $\tilde{r} = r + r'$ and $\tilde{u} = u - ar' + (b/z)u'$.

Ciphertext Query. The adversary $\mathcal{A}$ makes a ciphertext query for DFA $M = (Q, \mathcal{T}, q_0, q_{n-1})$. $\mathcal{B}$ creates the key for M as follows. We use the same notation as in the proof of Theorem 7, which we recall as follows. Denote by $\boldsymbol{w}^\star_i$ the vector formed by the last $\ell^\star - i$ symbol of $\boldsymbol{w}^\star$. That is $\boldsymbol{w}^\star_i = (w^\star_{i+1}, \ldots, w^\star_{\ell^\star})$. Hence $\boldsymbol{w}^\star_0 = \boldsymbol{w}^\star$ and $\boldsymbol{w}^\star_{\ell^\star}$ is the empty string. For $q_k \in \{q_0, \ldots, q_{n-1}\} = Q$, let M_k be the same DFA as M except that the start state is set to q_k. Then, for each $q_k \in Q$ we define $V_k = \{i \in [0, \ell^\star] \mid R(M_k, \boldsymbol{w}^\star_i) = 1\}$. From this and the query restriction that $R(M, w^\star) = 0$, we have $0 \notin V_0$. Since there is only one accept state q_{n-1} (due to the WLOG condition) and this state has no out-going transition, we have $V_{n-1} = \{\ell^\star\}$. We denote $V^{+1}_x = \{i + 1 \mid i \in V_x\}$.

(Programming Randomness in Ciphertext). $\mathcal{B}$ implicitly defines

$$s = -b/z$$

$$v = b, \tag{52}$$

$$s_0 = b\Big(\sum_{i \in V_0} d_{\ell^\star+1-i}\Big), \tag{53}$$

$$\forall_{q_x \in Q \smallsetminus \{q_{n-1}\}} \quad u_x = \sum_{i \in V_x} a^{\ell^\star+1-i} b. \tag{54}$$

We note that, from the identity $u_{n-1} = \phi_2 v$, we have $u_{n-1} = ab = \sum_{i \in V_{n-1}} a^{\ell^\star+1-i} b$, due to $V_{n-1} = \{\ell^\star\}$. Hence Equation (20) indeed holds for all $q_x \in Q$. Finally, $\mathcal{B}$ implicitly defines

$$\forall_{t \in [1,m]} \quad s_t = b\Big(- \sum_{i \in V^{+1}_{x_t}} a^{\ell^\star+1-i} f - \sum_{i \in V^{+1}_{x_t} \smallsetminus V_{y_t}} d_{\ell^\star+1-i}/(\sigma_t - w^\star_i) + \sum_{i \in V_{y_t} \smallsetminus V^{+1}_{x_t}} d_{\ell^\star+1-i}/(\sigma_t - w^\star_i) \Big). \tag{55}$$

We note that the programming in Eq. (52),(53),(54),(55) are exactly the same as in Eq. (17),(18),(20),(21) in the proof of Theorem 7, with the proper notation replacement, *i.e.*, variable s_0, v, s_t, u_x here correspond to variable r_0, r, r_t, u_x, respectively.

(Cancellation in Ciphertext). The cancellation in key here is exactly the same as that in ciphertext in the proof of Theorem 7 (where we replace the notation for variables properly), with one additional cancellation as follows.

- Canceling g^{ab} in $g^{s\phi + v\phi_1}$ by

$$g^{(-b/z)(az)} \text{ from } g^{(s)(\phi)}, \qquad\qquad g^{(b)(a)} \text{ from } g^{(v)(\phi_1)}$$

(Simulating Ciphertext). The simulation for key here is exactly the same as that for the corresponding terms in ciphertext in the proof of Theorem 7. That is, we simulate as in Eq.(24)-(29), with the proper notation replacement. The additional terms are simulated as follows.

$$g^{s\phi + v\phi_1} = (g^{b/z})^{-\phi'} (g^b)^{\phi'_1}$$

$$g^s = (g^{b/z})^{-1}$$

$$g^{s\eta} = (g^b)^{-1}.$$

(Re-randomizing Ciphertext). We can re-randomize $(v, s_0, \{s_t\}_{t \in [1,m]}, \{u_x\}_{q_x \in Q^\star})$ since $\mathcal{B}$ knows $g^{\phi_1}, g^{h_0}, \ldots, g^{h_4}$. This makes these randomness independent from s.

Guess. $\mathcal{A}$ will eventually output a guess if $\alpha = 0$ or random, our simulator $\mathcal{B}$ just outputs the guess that $\tau = 0$ and τ is random respectively. Hence, the advantage of $\mathcal{B}$ winning the $\ell^\star$-EDHE1-Dual game is exactly the same as the advantage of $\mathcal{A}$ winning the co-selective master-key hiding game. □

E Generic Security of the Assumptions

Definition 11 (**Expanded Diffie-Hellman Exponent Assumption in Subgroup**)**.** A (M,Y)-Expanded Diffie-Hellman Exponent Assumption in Subgroup (EDHE) is parameterized by a matrix $M \in \mathbb{Z}^{r\times k}$ and a vector $Y \in \mathbb{Z}^{1\times k}$. Given (M,Y) and a bilinear group generator $\mathcal{G}$ we define the following distribution:

$$(\mathbb{G}, \mathbb{G}_T, e, N, p_1, \dots, p_t) \xleftarrow{\$} \mathcal{G}(\lambda); \qquad g \xleftarrow{\$} \mathbb{G}_{p_1}; \dots; g_{p_t} \xleftarrow{\$} \mathbb{G}_{p_t}; \qquad x_1, \dots, x_k \xleftarrow{\$} \mathbb{Z}_p;$$
$$\forall_{i\in[1,r]}\ W_i := \prod_{j\in[1,k]} (x_j)^{M_{i,j}}; \qquad D := (\mathbb{G}, \mathbb{G}_T, e, N, g, g_{p_2}, \dots, g_{p_t}, g^{W_1}, \dots, g^{W_r});$$
$$Z := \prod_{j\in[1,k]} (x_j)^{Y_j}; \qquad T_1 := g^Z; \qquad T_2 \xleftarrow{\$} \mathbb{G}_{p_1}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking this assumption to be:

$$\mathsf{Adv}_{(M,Y)\text{-}\mathsf{EDHE},\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D,T_1)] - \Pr[\mathcal{A}(D,T_2)]|.$$

We say that $\mathcal{G}$ satisfies (M,Y)-EDHE assumption if $\mathsf{Adv}_{(M,Y)\text{-}\mathsf{EDHE},\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any probabilistic polynomial-time algorithm $\mathcal{A}$.

The following proposition follows the methodology of Boneh, Boyen, and Goh [5] and is implicitly used in [26, 38].

Proposition 42. *The (M,Y)-EDHE assumption is secure in the generic group model if the followings hold:*
1. *The integers $r, k, \forall_{i\in[1,r]}|M_{i,j}|, \forall_{j\in[1,k]}|Y_j|$ are $\mathcal{O}(\mathsf{poly}(\lambda))$.*
2. *For all $u, v, w \in [1,r]$ we have $Y + M_u \neq M_v + M_w$ and $2Y \neq M_v + M_w$, where M_i denotes the i-th row of M.*

Terminology. In what follows, we will argue that specific assumptions can be considered as (M,Y)-EDHE assumption where the matrix M and the vector Y are determined by that assumption. We will write M, Y explicitly in a table. In the table, each column is named by the corresponding variable. Each row corresponds to each W_i element given in the assumption and all the vectors except the last one together establish the matrix M, while the last vector depicts Y and corresponds to the target element in the assumption. We denote $(x_{@i})$ as the vector with all entries being 0 element except the i-th entry which is x and $(x_{@i}, y_{@j})$ as the vector with all entries being 0 element except the i-th entry which is x and j-th entry which is y.

Lemma 43. *The ℓ-EDHE1 assumption (Definition 3) is secure in the generic group model.*

Proof. The ℓ-EDHE1 assumption can be considered as (M,Y)-EDHE assumption where the matrix M and the vector Y are depicted in Table 3, and where we use variables $a, b, c, d_1, \dots, d_{\ell+1}, f, z$ instead of $x_1, \dots, x_{\ell+5}$ as in Definition 11. The terminology of the table is explained as above. We first observe that the first requirement holds since $\ell = \mathcal{O}(\mathsf{poly}(\lambda))$. We now prove the second

Table 3: The matrix representation of the ℓ-$\mathsf{EDHE1}$ assumption

Type	Terms	Range	a	b	c	d_1	d_2	$\cdots$	$d_{\ell+1}$	f	z
1	g		0	0	0	0	0	$\cdots$	0	0	0
2	g^a		1	0	0	0	0	$\cdots$	0	0	0
3	g^b		0	1	0	0	0	$\cdots$	0	0	0
4	$g^{a/f}$		1	0	0	0	0	$\cdots$	0	-1	0
5	$g^{1/f}$		0	0	0	0	0	$\cdots$	0	-1	0
6	$g^{a^\ell c/z}$		ℓ	0	1	0	0	$\cdots$	0	0	-1
7	$g^{a^i bf}$	$i \in [1, \ell+1]$	i	1	0	0	0	$\cdots$	0	1	0
8	$g^{a^i c}$	$i \in [0, \ell]$	i	0	1	0	0	$\cdots$	0	0	0
9	g^{a^i/d_i}	$i \in [1, \ell+1]$	i	0	0	$-1_{@i}$				0	0
10	g^{bd_i}	$i \in [0, \ell]$	0	1	0	$1_{@i}$				0	0
11	$g^{bd_i/f}$	$i \in [0, \ell]$	0	1	0	$1_{@i}$				-1	0
12	$g^{abd_i/f}$	$i \in [0, \ell]$	1	1	0	$1_{@i}$				-1	0
13	$g^{a^i c/d_j}$	$i \in [1, 2\ell+1], i \neq \ell+1, j \in [1, \ell+1]$	i	0	1	$-1_{@j}$				0	0
14	$g^{a^i bf/d_j}$	$i \in [2, 2\ell+2], j \in [1, \ell+1]$	i	1	0	$-1_{@j}$				1	0
15	$g^{a^i bd_j/d_i}$	$i, j \in [1, \ell+1], i \neq j$	i	1	0	$1_{@j}, -1_{@i}$				0	0
	Target										
$\star$	g^{abz}		1	1	0	0	0	$\cdots$	0	0	1

requirement. We denote by $\mathsf{v}_{x,i,j}$ the row of type x with specified i, j in the range if there is any for that type. We also denote by S_x the set of all row indexes of type x ranged in its specified condition. We first observe that $2\mathsf{v}_\star$ contains 2 in the column z, but for any v, w, $\mathsf{v}_v + \mathsf{v}_w$ contains at most 0 in the that column, hence $2\mathsf{v}_\star \neq \mathsf{v}_v + \mathsf{v}_w$ for any v, w. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all u, v, w. We observe that $\mathsf{v}_\star + \mathsf{v}_u$ for $u \neq 6$ contains 1 in column z. Hence by the same reason, $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all $u \neq 6, v, w$. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_6 = (\ell+1, 1, 1, 0, \ldots, 0) \neq \mathsf{v}_v + \mathsf{v}_w$ for all v, w. We categorize into the following cases. For a vector X and column q, we denote $[X]_q$ the entry in X at column q.

- $v = 6$ or $w = 6$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_z \leq -1$ but $[\mathsf{v}_\star + \mathsf{v}_6]_z = 0$.
- $v \notin S_8 \cup S_{13} \cup \{6\}$ and $w \notin S_8 \cup S_{13} \cup \{6\}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 0$ but $[\mathsf{v}_\star + \mathsf{v}_6]_c = 1$.
- $v \in S_8 \cup S_{13}$ and $w \in S_8 \cup S_{13}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 2$ but $[\mathsf{v}_\star + \mathsf{v}_6]_c = 1$.
- $v \in S_8$ and $w \notin S_8 \cup S_{13} \cup \{6\}$. We further categorize:
  - $w = 3$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \leq \ell$ but $[\mathsf{v}_\star + \mathsf{v}_6]_a = \ell + 1$.
  - $w \in S_7$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_f = 1$ but $[\mathsf{v}_\star + \mathsf{v}_6]_f = 0$.
  - $w \in \{2, 4, 5\} \cup S_9$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_b = 0$ but $[\mathsf{v}_\star + \mathsf{v}_6]_b = 1$.
  - $w \in S_{10} \cup S_{11} \cup S_{12} \cup S_{14} \cup S_{15}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_6]_{d_j} = 0$ for all j.
- $v \in S_{13}$ and $w \notin S_8 \cup S_{13} \cup \{6\}$. We further categorize:
  - $w \in S_{10}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \neq \ell + 1$ but $[\mathsf{v}_\star + \mathsf{v}_6]_a = \ell + 1$.
  - $w \in S_{11} \cup S_{12}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_f = -1$ but $[\mathsf{v}_\star + \mathsf{v}_6]_f = 0$.

- $w \notin S_{10} \cup S_{11} \cup S_{12}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \leq -1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_6]_{d_j} = 0$ for all j.

- $v \notin S_8 \cup S_{13} \cup \{6\}$ and $w \in S_8 \cup S_{13}$. This is the same as the two previous cases by exchanging v, w.

This concludes all cases. □

Lemma 44. *The ℓ-EDHE1-Dual assumption (Definition 10) is secure in the generic group model.*

Proof. The ℓ-EDHE1-Dual assumption is different from the ℓ-EDHE1 assumption only at the type 6 and the target, where they are changed to $g^{b/z}$ and $g^{a^{\ell+1}cz}$ respectively. By the same argument as the previous proof, it can be shown that $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all $u \neq 6, v, w$. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_6 \neq \mathsf{v}_v + \mathsf{v}_w$ for all v, w. But the modification gives the same $\mathsf{v}_\star + \mathsf{v}_6 = (\ell+1, 1, 1, 0, \ldots, 0)$. Hence, the proof follows the previous proof from this point on. □

Table 4: The matrix representation of the (n, m)-EDHE2 assumption

Type	Terms	Range	a	b	c	d_1	d_2	$\cdots$	d_m	z
1	g		0	0	0	0	0	$\cdots$	0	0
2	g^a		1	0	0	0	0	$\cdots$	0	0
3	g^b		0	1	0	0	0	$\cdots$	0	0
4	$g^{a^{n-1}c/z}$		$n-1$	0	1	0	0	$\cdots$	0	-1
5	g^{a^ic}	$i \in [0, n-1]$	i	0	1	0	0	$\cdots$	0	0
6	g^{d_j}	$j \in [1, m]$	0	0	0	$1_{@j}$				0
7	g^{a^i/d_j^2}	$i \in [1, n], j \in [1, m]$	i	0	0	$-2_{@j}$				0
8	g^{a^i/d_j^6}	$i \in [1, n], j \in [1, m]$	i	0	0	$-6_{@j}$				0
9	g^{a^ib/d_j}	$i \in [1, n], j \in [1, m]$	i	1	0	$-1_{@j}$				0
10	$g^{a^id_j/d_{j'}^2}$	$i \in [1, n], j, j' \in [1, m], j \neq j'$	i	0	0	$1_{@j}, -2_{@j'}$				0
11	$g^{a^id_j/d_{j'}^6}$	$i \in [1, n], j, j' \in [1, m], j \neq j'$	i	0	0	$1_{@j}, -6_{@j'}$				0
12	$g^{a^ibd_j/d_{j'}}$	$i \in [1, n], j, j' \in [1, m], j \neq j'$	i	1	0	$1_{@j}, -1_{@j'}$				0
13	$g^{a^ibcd_j}$	$i \in [0, n-1], j \in [1, m]$	i	1	1	$1_{@j}$				0
14	$g^{a^ibcd_j^5}$	$i \in [0, n], j \in [1, m]$	i	1	1	$5_{@j}$				0
15	$g^{a^ibcd_j/d_{j'}^2}$	$i \in [1, 2n-1], j, j' \in [1, m], j \neq j'$	i	1	1	$1_{@j}, -2_{@j'}$				0
16	$g^{a^ibcd_j^5/d_{j'}^6}$	$i \in [1, 2n-1], j, j' \in [1, m], j \neq j'$	i	1	1	$5_{@j}, -6_{@j'}$				0
17	g^{a^ibc/d_j}	$i \in [1, 2n-1], i \neq n, j \in [1, m]$	i	1	1	$-1_{@j}$				0
18	g^{a^ic/d_j^2}	$i \in [1, 2n-1], j \in [1, m]$	i	0	1	$-2_{@j}$				0
19	g^{a^ic/d_j^6}	$i \in [1, 2n-1], j \in [1, m]$	i	0	1	$-6_{@j}$				0
20	$g^{a^ib^2cd_j/d_{j'}}$	$i \in [1, 2n-1], j, j' \in [1, m]$	i	2	1	$1_{@j}, -1_{@j'}$				0
21	$g^{a^ib^2cd_j^5/d_{j'}}$	$i \in [1, 2n-1], j, j' \in [1, m]$	i	2	1	$5_{@j}, -1_{@j'}$				0
22	$g^{a^ibcd_j/d_{j'}^6}$	$i \in [1, 2n-1], j, j' \in [1, m]$	i	1	1	$1_{@j}, -6_{@j'}$				0
23	$g^{a^ibcd_j^5/d_{j'}^2}$	$i \in [1, 2n-1], j, j' \in [1, m]$	i	1	1	$5_{@j}, -2_{@j'}$				0
	Target									
$\star$	g^{abz}		1	1	0	0	0	$\cdots$	0	1

Lemma 45. *The (n, m)-$\mathsf{EDHE2}$ assumption (Definition 4) is secure in the generic group model.*

Proof. The (n, m)-$\mathsf{EDHE2}$ assumption can be considered as (M, Y)-EDHE assumption where the matrix M and the vector Y are depicted in Table 4, and where we use variables $a, b, c, d_1, \ldots, d_m, z$ instead of $x_1, \ldots, x_{m+4}$ as in Definition 11. The first requirement holds since $n, m, = \mathcal{O}(\mathsf{poly}(\lambda))$. We now prove the second requirement. We denote by $\mathsf{v}_{x,i,j}$ and S_x similarly as in the previous proof. We first observe that $2\mathsf{v}_\star$ contains 2 in the column z, but for any v, w, $\mathsf{v}_v + \mathsf{v}_w$ contains at most 0 in the that column, hence $2\mathsf{v}_\star \neq \mathsf{v}_v + \mathsf{v}_w$ for any v, w. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all u, v, w. We observe that $\mathsf{v}_\star + \mathsf{v}_u$ for $u \neq 4$ contains 1 in column z. Hence by the same reason, $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all $u \neq 4, v, w$. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_4 = (n, 1, 1, 1, 0, \ldots, 0) \neq \mathsf{v}_v + \mathsf{v}_w$ for all v, w. We categorize into the following cases. For a vector X and column q, we denote $[X]_q$ the entry in X at column q. We first consider the following three cases.

- $v = 4$ or $w = 4$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_z \leq -1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_z = 0$.
- $v \in S_{20} \cup S_{21}$ or $w \in S_{20} \cup S_{21}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_b \geq 2$ but $[\mathsf{v}_\star + \mathsf{v}_4]_b = 1$.
- $v \in S_8 \cup S_{11} \cup S_{16} \cup S_{19}$ or $w \in S_8 \cup S_{11} \cup S_{16} \cup S_{19}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \leq -1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j. This is since $[\mathsf{v}_v]_{d_j} = -6$ for some j and $[\mathsf{v}_w]_{d_j} \leq 5$ for all j.
- $v \in S_{23}$ or $w \in S_{23}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j. This is due to the following. WLOG, we assume $v \in S_{23}$ (and w can be any) and write $v = (23, i, j, j')$. We categorize as:
  - If $j = j'$, $[\mathsf{v}_v]_{d_j} = 3$. But for all j, $[\mathsf{v}_w]_{d_j} \neq -3$.
  - If $j \neq j'$, $([\mathsf{v}_v]_{d_j}, [\mathsf{v}_v]_{d'_j}) = (5, -2)$. But for all j, j', $([\mathsf{v}_w]_{d_j}, [\mathsf{v}_w]_{d'_j}) \neq (-5, 2)$.
- $v \in S_{14}$ or $w \in S_{14}$. WLOG, we assume $v \in S_{14}$. Then we categorize as:
  - $w \in S_{22}$. Then $[\mathsf{v}_v + \mathsf{v}_w]_b \geq 2$ but $[\mathsf{v}_\star + \mathsf{v}_4]_b = 1$.
  - $w \notin S_{22}$. Then $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j. This is since $[\mathsf{v}_v]_{d_j} = 5$ for some j and $[\mathsf{v}_w]_{d_j} \neq -5$ for all j.

From now, we can assume $v, w \notin \{4\} \cup S_8 \cup S_{11} \cup S_{14} \cup S_{16} \cup S_{19} \cup S_{20} \cup S_{21} \cup S_{23}$. We then consider the following case:

- $v \in S_7 \cup S_{10} \cup S_{15} \cup S_{18} \cup S_{22}$ or $w \in S_7 \cup S_{10} \cup S_{15} \cup S_{18} \cup S_{22}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \leq -1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j. This is since $[\mathsf{v}_v]_{d_j} \leq -2$ for some j and $[\mathsf{v}_w]_{d_j} \leq 1$ for all j.

From now, we can assume also $v, w \notin S_7 \cup S_{10} \cup S_{15} \cup S_{18} \cup S_{22}$. We further categorize as:

- $v \in S_5 \cup S_{13} \cup S_{17}$ and $w \in S_5 \cup S_{13} \cup S_{17}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 2$ but $[\mathsf{v}_\star + \mathsf{v}_4]_c = 1$.
- $v \notin S_5 \cup S_{13} \cup S_{17}$ and $w \notin S_5 \cup S_{13} \cup S_{17}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 0$ but $[\mathsf{v}_\star + \mathsf{v}_4]_c = 1$.
- $v \in S_5$ and $w \notin S_5 \cup S_{13} \cup S_{17}$. We further categorize:
  - $w = 3$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \leq n - 1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_a = n$.
  - $w = 2$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_b = 0$ but $[\mathsf{v}_\star + \mathsf{v}_4]_b = 1$.
  - $w \notin \{2, 3\}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j.
- $v \in S_{13}$ and $w \notin S_5 \cup S_{13} \cup S_{17}$. We further categorize:

  - $w = 2$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} = 1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j.
  - $w \in S_6$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} \geq 1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j.
  - $w \notin \{2\} \cup S_6$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_b > 1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_b = 1$.

- $v \in S_{17}$ and $w \notin S_5 \cup S_{13} \cup S_{17}$. We further categorize:
  - $w = 2$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{d_j} = -1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = 0$ for all j.
  - $w \in S_6$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \neq n$ but $[\mathsf{v}_\star + \mathsf{v}_4]_{d_j} = n$.
  - $w \notin \{2\} \cup S_6$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_b > 1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_b = 1$.
- $v \notin S_5 \cup S_{13} \cup S_{17}$ and $w \in S_5 \cup S_{13} \cup S_{17}$. This is the same as the 3 previous cases by exchanging v, w.

This concludes all cases. □

Lemma 46. *The (n, m)-EDHE2-Dual assumption (Definition 9) is secure in the generic group model.*

Proof. The (n, m)-EDHE2-Dual assumption is different from the (n, m)-EDHE2 assumption only at the type 4 and the target, where they are changed to $g^{b/z}$ and $g^{a^n cz}$ respectively. By the same argument as the previous proof, it can be shown that $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all $u \neq 4, v, w$. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_4 \neq \mathsf{v}_v + \mathsf{v}_w$ for all v, w. But the modification gives the same $\mathsf{v}_\star + \mathsf{v}_4 = (n, 1, 1, 0, \ldots, 0)$. Hence, the proof follows the previous proof from this point on. □

Table 5: The matrix representation of the (n, t)-EDHE3 assumption

Type	Terms	Range	a	c	b_1	b_2	$\cdots$	b_t	z
1	g		0	0	0	0	$\cdots$	0	0
2	g^a		1	0	0	0	$\cdots$	0	0
3	g^c		0	1	0	0	$\cdots$	0	0
4	$g^{c/z}$		0	1	0	0	$\cdots$	0	-1
5	g^{b_j}	$j \in [1, t]$	0	0	$1_{@j}$				0
6	g^{a^i/b_j^2}	$i \in [1, n+1], j \in [1, t]$	i	0	$-2_{@j}$				0
7	$g^{a^i b_j / b_{j'}^2}$	$i \in [1, n], j, j' \in [1, t], j \neq j'$	1	0	$1_{@j}, -2_{@j'}$				0
8	$g^{a^n c b_j / b_{j'}}$	$j, j' \in [1, t], j \neq j'$	n	1	$1_{@j}, -1_{@j'}$				0
9	$g^{a^i c b_j}$	$i \in [1, 2n], j \in [1, t]$	i	1	$1_{@j}$				0
10	$g^{a^i c / b_j}$	$i \in [1, 2n], i \neq n+1, j \in [1, t]$	i	1	$-1_{@j}$				0
11	$g^{a^i c b_j / b_{j'}^2}$	$i \in [1, 2n], j, j' \in [1, t], j \neq j'$	i	1	$1_{@j}, -2_{@j'}$				0
12	$g^{a^i c^2 b_j / b_{j'}}$	$i \in [n+1, 2n], j, j' \in [1, t]$	i	2	$1_{@j}, -1_{@j'}$				0
	Target								
$\star$	$g^{a^{n+1} z}$		$n+1$	0	0	0	0	$\cdots$	1

Lemma 47. *The (n, t)-EDHE3 assumption (Definition 6) is secure in the generic group model.*

Proof. The (n,t)-$\mathsf{EDHE3}$ assumption can be considered as (M,Y)-EDHE assumption where the matrix M and the vector Y are depicted in Table 5, and where we use variables $a, c, b_1, \dots, b_t, z$. The first requirement holds since $n, t = \mathcal{O}(\mathsf{poly}(\lambda))$. We now prove the second requirement. We denote by $\mathsf{v}_{x,i,j}$ and S_x similarly as in the previous proof. We first observe that $2\mathsf{v}_\star$ contains 2 in the column z, but for any v, w, $\mathsf{v}_v + \mathsf{v}_w$ contains at most 0 in the that column, hence $2\mathsf{v}_\star \neq \mathsf{v}_v + \mathsf{v}_w$ for any v, w. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all u, v, w. We observe that $\mathsf{v}_\star + \mathsf{v}_u$ for $u \neq 4$ contains 1 in column z. Hence by the same reason, $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all $u \neq 4, v, w$. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_4 = (n+1, 1, 0, \dots, 0) \neq \mathsf{v}_v + \mathsf{v}_w$ for all v, w. We categorize into the following cases. For a vector X and column q, we denote $[X]_q$ the entry in X at column q. We first consider the following three cases.

- $v = 4$ or $w = 4$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_z \leq -1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_z = 0$.
- $v \in S_6 \cup S_7 \cup S_{11}$ or $w \in S_6 \cup S_7 \cup S_{11}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{b_j} \leq -1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{b_j} = 0$ for all j. This is since $[\mathsf{v}_v]_{b_j} = -2$ for some j and $[\mathsf{v}_w]_{b_j} \leq 1$ for all j.
- $v \in S_{12}$ or $w \in S_{12}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c \geq 2$ but $[\mathsf{v}_\star + \mathsf{v}_4]_c = 1$.

From now, we can assume $v, w \notin \{4\} \cup S_6 \cup S_7 \cup S_{11} \cup S_{12}$. We further categorize as:

- $v \in \{3\} \cup S_8 \cup S_9 \cup S_{10}$ and $w \in \{3\} \cup S_8 \cup S_9 \cup S_{10}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 2$ but $[\mathsf{v}_\star + \mathsf{v}_4]_c = 1$.
- $v \notin \{3\} \cup S_8 \cup S_9 \cup S_{10}$ and $w \notin \{3\} \cup S_8 \cup S_9 \cup S_{10}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 0$ but $[\mathsf{v}_\star + \mathsf{v}_4]_c = 1$.
- $v \in \{3\} \cup S_8 \cup S_9 \cup S_{10}$ and $w \notin \{3\} \cup S_8 \cup S_9 \cup S_{10}$. Hence $w \in \{2, 5\}$. We further categorize as:
  - $v = 3$ (and $w \in \{2, 5\}$). Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \neq n+1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_a = n+1$.
  - $v \in S_8 \cup S_9 \cup S_{10}$ and $w = 2$. Then $[\mathsf{v}_v + \mathsf{v}_w]_{b_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{b_j} = 0$ for all j.
  - $v \in S_8 \cup S_9$ and $w = 5$. Then $[\mathsf{v}_v + \mathsf{v}_w]_{b_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_4]_{b_j} = 0$ for all j.
  - $v \in S_{10}$ and $w = 5$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \neq n+1$ but $[\mathsf{v}_\star + \mathsf{v}_4]_a = n+1$.
- $v \notin \{3\} \cup S_8 \cup S_9 \cup S_{10}$ and $w \in \{3\} \cup S_8 \cup S_9 \cup S_{10}$. This is the same as the previous case by exchanging v, w.

This concludes all cases. □

Lemma 48. *The (n,m,k)-$\mathsf{EDHE4}$ assumption (Definition 7) is secure in the generic group model.*

Proof. The (n,m,k)-$\mathsf{EDHE4}$ assumption can be considered as (M,Y)-EDHE assumption where the matrix M and the vector Y are depicted in Table 6, and where we use variables $a, x, c, b_1, \dots, b_m, z$. The first requirement holds since $n, m, k = \mathcal{O}(\mathsf{poly}(\lambda))$. We now prove the second requirement. We denote by $\mathsf{v}_{x,i,j}$ and S_x similarly as in the previous proof. We first observe that $2\mathsf{v}_\star$ contains 2 in the column z, but for any v, w, $\mathsf{v}_v + \mathsf{v}_w$ contains at most 0 in the that column, hence $2\mathsf{v}_\star \neq \mathsf{v}_v + \mathsf{v}_w$ for any v, w. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all u, v, w. We observe that $\mathsf{v}_\star + \mathsf{v}_u$ for $u \neq 3$ contains 1 in column z. Hence by the same reason, $\mathsf{v}_\star + \mathsf{v}_u \neq \mathsf{v}_v + \mathsf{v}_w$ for all $u \neq 3, v, w$. It remains to prove that $\mathsf{v}_\star + \mathsf{v}_3 = (n+1, k+1, 1, 0, \dots, 0) \neq \mathsf{v}_v + \mathsf{v}_w$ for all v, w. We categorize into the following cases. For a vector X and column q, we denote $[X]_q$ the entry in X at column q. We first consider the following two cases.

- $v = 3$ or $w = 3$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_z \leq -1$ but $[\mathsf{v}_\star + \mathsf{v}_3]_z = 0$.

Table 6: The matrix representation of the (n, m, k)-$\mathsf{EDHE4}$ assumption

Type	Terms	Range	a	x	c	b_1	b_2	$\cdots$	b_m	z
1	g		0	0	0	0	0	$\cdots$	0	0
2	g^c		0	0	1	0	0	$\cdots$	0	0
3	$g^{a^{n+1}x^k/z}$		$n+1$	k	0	0	0	$\cdots$	0	-1
4	$g^{a^{n+1}x^j}$	$j \in [1, k]$	$n+1$	j	0	0	0	$\cdots$	0	0
5	g^{x^j}	$j \in [1, k]$	0	j	0	0	0	$\cdots$	0	0
6	$g^{a^i x^j / b_\iota^2}$	$i \in [1, n], j \in [1, k], \iota \in [1, m]$	i	j	0	$-2_{@\iota}$				0
7	g^{cb_ι}	$\iota \in [1, m]$	0	0	1	$1_{@\iota}$				0
8	$g^{a^i x^j b_\iota}$	$i \in [1, n], j \in [1, k], \iota \in [1, m]$	i	j	0	$1_{@\iota}$				0
9	$g^{a^{n+1}x^j cb_\iota / b_{\iota'}}$	$j \in [1, k], \iota, \iota' \in [1, m], \iota \neq \iota'$	$n+1$	j	1	$1_{@\iota}, -1_{@\iota'}$				0
10	$g^{a^i x^j cb_\iota / b_{\iota'}^2}$	$i \in [1, n], j \in [1, k], \iota, \iota' \in [1, m], \iota \neq \iota'$	i	j	1	$1_{@\iota}, -2_{@\iota'}$				0
11	$g^{a^i x^j b_\iota / b_{\iota'}^2}$	$i \in [1, 2n], j \in [1, 2k], \iota, \iota' \in [1, m], \iota \neq \iota'$	i	j	0	$1_{@\iota}, -2_{@\iota'}$				0
12	$g^{a^i x^j / b_\iota}$	$i \in [1, 2n], j \in [1, 2k], \iota, \in [1, m]$, $(i, j) \neq (n+1, k+1)$	i	j	0	$-1_{@\iota}$				0
	Target									
$\star$	g^{xcz}		0	1	1	0	0	$\cdots$	0	1

- $v \in S_6 \cup S_{10} \cup S_{11}$ or $w \in S_6 \cup S_{10} \cup S_{11}$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{b_j} \leq -1$ for some j but $[\mathsf{v}_\star + \mathsf{v}_3]_{b_j} = 0$ for all j. This is since $[\mathsf{v}_v]_{b_j} = -2$ for some j and $[\mathsf{v}_w]_{b_j} \leq 1$ for all j.

From now, we can assume $v, w \notin \{3\} \cup S_6 \cup S_{10} \cup S_{11}$. We further categorize as:

- $v \in \{2\} \cup S_7 \cup S_9$ and $w \in \{2\} \cup S_7 \cup S_9$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 2$ but $[\mathsf{v}_\star + \mathsf{v}_3]_c = 1$.
- $v \notin \{2\} \cup S_7 \cup S_9$ and $w \notin \{2\} \cup S_7 \cup S_9$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_c = 0$ but $[\mathsf{v}_\star + \mathsf{v}_3]_c = 1$.
- $v \in \{2\} \cup S_7$ and $w \notin \{2\} \cup S_7 \cup S_9$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_a \neq n + 1$ or $[\mathsf{v}_v + \mathsf{v}_w]_x \neq k + 1$ while $[\mathsf{v}_\star + \mathsf{v}_3]_a = n + 1$ and $[\mathsf{v}_\star + \mathsf{v}_3]_x = k + 1$.
- $v \in S_9$ and $w \notin \{2\} \cup S_7 \cup S_9$. Then, $[\mathsf{v}_v + \mathsf{v}_w]_{b_j} \neq 0$ for some j but $[\mathsf{v}_\star + \mathsf{v}_3]_{b_j} = 0$ for all j.
- $v \notin S_5 \cup S_{11} \cup S_{13}$ and $w \in S_5 \cup S_{11} \cup S_{13}$. This is the same as the 2 previous cases by exchanging v, w.

This concludes all cases. □

References

[1] N. Attrapadung, B. Libert. Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In *PKC'10*, *LNCS* 6056 pp. 384–402, 2010.

[2] N. Attrapadung, B. Libert, E. Panafieu. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts In *PKC'11*, *LNCS* 6571, pp. 90–108.

[3] J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. *IEEE Symposium on Security and Privacy'07*, pp. 321-334, 2007.

[4] D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.

[5] D. Boneh, X. Boyen, E.-J. Goh. Hierarchical Identity-Based encryption with Constant Size Ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.

[6] D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt'08*, *LNCS* 5350, pp. 455–470, 2008.

[7] D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC'11*, *LNCS* 6597, pp. 253–273, 2011.

[8] J. Chen, H. Wee. Fully, (Almost) Tightly Secure IBE from Standard Assumptions. In *Crypto'13*, *LNCS* 8043, pp. 435–460, 2013.

[9] C. Chen, J. Chen, H. W. Lim, Z. Zhang, D. Feng, S. Ling, H. Wang. Fully Secure Attribute-Based Systems with Short Ciphertexts/Signatures and Threshold Access Structures. In *CT-RSA'13*, *LNCS* 7779, pp. 50–67, 2013.

[10] C. Chen, Z. Zhang, D. Feng. Fully Secure Doubly-Spatial Encryption under Simple Assumptions. In *ProvSec 2012*, *LNCS* 7496, pp. 253–263, 2012.

[11] P. Erd̈os, P. Frankel, Z. Furedi. Families of finite sets in which no set is covered by the union of r others. In *Israeli Journal of Mathematics*, 51, pp. 79âĂŞ-89, 1985.

[12] S. Garg, C. Gentry, S. Halevi. Candidate multilinear maps from ideal lattices In *Eurocrypt'13*, *LNCS* 7881, pp. 1–17, 2013.

[13] S. Garg, C. Gentry, S. Halevi, A. Sahai, B. Waters. Attribute-based encryption for circuits from multilinear maps. In *Crypto'13*, *LNCS* 8043, pp. 479-499, 2013.

[14] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, B. Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits. In *FOCS'13*, pp. 40–49, 2013.

[15] S. Goldwasser, Y. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In *STOC'13*, pp. 555–564.

[16] S. Goldwasser, Y. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich. How to run Turing machines on encrypted data. In *Crypto'13*, *LNCS* 8043, pp. 536-553, 2013.

[17] S. Gorbunov, V. Vaikuntanathan, H. Wee. Attribute-based encryption for circuits. In *STOC'13*, pp. 545–554, 2013.

[18] V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, pp. 89–98, 2006.

[19] M. Hamburg. Spatial Encryption. Cryptology ePrint Archive: Report 2011/389.

[20] J. Katz, A. Sahai, B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *Eurocrypt'08*, *LNCS* 4965, pp. 146-162.

[21] R. Kumar, S. Rajagopalan, A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In *Crypto'99*, *LNCS* 1666, pp. 609–623, 1999.

[22] A. Lewko Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In *Eurocrypt'12*, *LNCS* 7237, pp. 318–335, 2012.

[23] A. Lewko, B. Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC'10*, *LNCS* 5978, pp. 455–479, 2010.

[24] A. Lewko, B. Waters. Decentralizing Attribute-Based Encryption In *Eurocrypt'11*, *LNCS* 6632, pp. 568–588, 2011.

[25] A. Lewko, B. Waters. Unbounded HIBE and Attribute-Based Encryption In *Eurocrypt'11*, *LNCS* 6632, pp. 547–567, 2011.

[26] A. Lewko, B. Waters. New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In *Crypto'12*, *LNCS*, pp. 180–198.

[27] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt'10*, *LNCS* 6110, pp. 62–91, 2010.

[28] M. Naor, O. Reingold. Number-Theoretic Constructions of Efficient Pseudo-Random Functions. In *Journal of ACM*, 51(2), pp. 231–262, 2004.

[29] Y. Rouselakis, B. Waters Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM CCS'13*, pp. 463–474, 2013.

[30] W.Peterson, E. Weldon. Error-Correcting Codes. Second Edition.

[31] A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt'05*, *LNCS* 3494, pp. 457–473, 2005.

[32] T. Okamoto, K. Takashima, Fully secure functional encryption with general relations from the decisional linear assumption. In *Crypto'10*, *LNCS* 6223, pp. 191-208.

[33] T. Okamoto, K. Takashima, Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In *Eurocrypt'12*, *LNCS* 7237, pp. 591-608, 2012.

[34] T. Okamoto, K. Takashima, Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In *Asiacrypt'12*, *LNCS* 7658, pp. 349-366, 2012.

[35] S.C. Ramanna. DFA-Based Functional Encryption: Adaptive Security from Dual System Encryption. Cryptology ePrint Archive: Report 2013/638.

[36] B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC'11*, *LNCS* 6571, pp. 53–70, 2011.

[37] B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto'09*, *LNCS* 5677, pp. 619–636, 2009.

[38] B. Waters. Functional Encryption for Regular Languages. In *Crypto'12*, *LNCS* 7417, pp. 218–235, 2012.

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

Contents