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

Nuttapong Attrapadung

National Institute of Advanced Industrial Science and Technology (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.

1 Introduction

Dual system encryption techniques introduced by Waters [33] 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) [33,20,22,19],

P.Q. Nguyen and E. Oswald (Eds.): EUROCRYPT 2014, LNCS 8441, pp. 557–577, 2014.

attribute-based encryption (ABE) for Boolean formulae [24,28,23], inner-product encryption [24,28,1,29,30], and spatial encryption [1,17].

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 [34], 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 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

[1] One explicit example is the fully secure HIBE of Lewko and Waters [20], 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.

is either perfectly master-key hiding or doubly selectively master-key hiding. Our construction is based on composite-order bilinear groups.

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 [34]. 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 [22,26] or small-universe [23] or restricted for multi-use of attributes [30].
- The first fully-secure key-policy ABE with constant-size ciphertexts. 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 ([34,26,2], 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 $O(q_{\mathrm{all}})$ 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 [20].

More Results. We also obtain some more results, which could not fit in the space here. These include a generic conversion for dual primitives (*i.e.,* key-policy to ciphertext-policy and vice-versa) for perfectly secure encoding, the first dual FE for regular languages, a unified treatment for existing FE schemes and improvements for ABE scheme of [24] (reducing key sizes to half for free, and a large-universe variant), a new primitive called key-policy over doubly spatial encryption, which unifies KP-ABE and (doubly) spatial encryption [17].

1.2 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 [3,6,16,18,27,32]. Recent results give very general FE primitives such as ABE or FE for circuits [15,11,13,12], and for Turing Machines [14], but most of them might still be considered as proofs of concept, since underlying cryptographic tools such as multilinear maps [10] seem still inefficient. Constructing fully secure ABE for circuits *without complexity leveraging* is an open problem.

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 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

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

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

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

where '$\cdot$' 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 [24]).

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

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*

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 [23,8]
$\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})}, g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}\right\}$	$,\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})}, g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}\right\}$	computational	[23]
$\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})}, \{g_2^{\boldsymbol{k}_i(0,\hat{\boldsymbol{r}}_i,\hat{\boldsymbol{h}})}\}_{i\in Q}\right\}$	$,\left\{g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})}, \{g_2^{\boldsymbol{k}_i(\hat{\alpha},\hat{\boldsymbol{r}}_i,\hat{\boldsymbol{h}})}\}_{i\in Q}\right\}$	computational	new

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 [23]). 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 $\boldsymbol{h} = \boldsymbol{h}' \bmod p_1$ and $\hat{\boldsymbol{h}} = \boldsymbol{h}' \bmod p_2$, 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 [23,8]) 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 [23] 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 $O(q_{\mathrm{all}})$ transition steps, hence results in $O(q_{\mathrm{all}})$ 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

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

[3] As a result, this also clarifies why [23] 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 [25] PRFs for their computational argument, which is different from ours.

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 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 $b \xleftarrow{\$} \{0,1\}$ 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 $\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) := |\Pr[b = b'] - \frac{1}{2}|$.

3.2 Definitions for Some Concrete Functional Encryption

FE for Regular Languages (DFA-based FE). In this primitive, 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 $\rho_0, \rho_1, \cdots, \rho_n \in Q$ such that $\rho_0 = q_0$, for $i = 1$ to ℓ we have $(\rho_{i-1}, \rho_i, w_i) \in \mathcal{T}$, and $\rho_\ell \in F$. 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. We note that it is wlog if we consider machines such that $|F| = 1$ (see the full version), and we will construct our scheme with this wlog condition.

Attribute Based Encryption 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 formulae 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. It 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: $\sum_{i \in I} \mu_i (A_i \boldsymbol{v}^\top) = s$.

3.3 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 [33,20] and the 3DH assumption in a subgroup [23] are given below.

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

1. *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* $T = T_2 \xleftarrow{\$} \mathbb{G}_{p_1}$.
2. *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* $T = T_2 \xleftarrow{\$} \mathbb{G}_{p_1p_3}$.
3. *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* $g_1, g_2, Z_3, g_1^\alpha Y_2, g_1^s W_2$, *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* $\mathsf{Adv}^{\mathsf{SD}i}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\boldsymbol{D}, T_1) = 1] - \Pr[\mathcal{A}(\boldsymbol{D}, T_2) = 1]|$, *where* $\boldsymbol{D}$ *denotes the given elements in each assumption excluding* T. *We say that the Assumption* i *holds for* $\mathcal{G}$ *if* $\mathsf{Adv}^{\mathsf{SD}i}_{\mathcal{A}}(\lambda)$ *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* $T \xleftarrow{\$} \mathbb{G}_{p_2}$.

Notation. In general, we treat a vector as a horizontal vector. For $g \in \mathbb{G}$ and $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{Z}^n$, we denote $g^{\boldsymbol{c}} = (g^{c_1}, \ldots, g^{c_n})$. Denote '$\cdot$' as the pairwise group operation on vectors. Consider $M \in \mathbb{Z}_N^{d \times n}$. We denote its transpose as $M^\top$. We denote by g^M the matrix in $\mathbb{G}^{d \times n}$ of which its (i, j) entry is $g^{M_{i,j}}$. 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.
- $\mathsf{Enc1}(X, N) \to \boldsymbol{k} = (k_1, \ldots, k_{m_1})$ and m_2. It takes as inputs $X \in \mathbb{X}_\kappa$, $N \in \mathbb{N}$, and outputs a sequence of polynomials $(k_z)_{z\in[1,m_1]}$ with coefficients in $\mathbb{Z}_N$, and $m_2 \in \mathbb{N}$. We require that each polynomial k_z is a *linear combination of monomials* $\alpha, r_i, r_i h_j$, where $\alpha, r_1, \ldots, r_{m_2}, h_1, \ldots, h_n$ are variables.
- $\mathsf{Enc2}(Y, N) \to \boldsymbol{c} = (c_1, \ldots, c_{w_1})$ and w_2. It takes as inputs $Y \in \mathbb{Y}_\kappa$, $N \in \mathbb{N}$, and outputs a sequence of polynomials $(c_z)_{z\in[1,w_1]}$ with coefficients in $\mathbb{Z}_N$, and $w_2 \in \mathbb{N}$. We require that each polynomial c_z is a *linear combination of monomials* $s, s_i, sh_j, s_i h_j$, where $s, s_1, \ldots, s_{w_2}, h_1, \ldots, h_n$ are variables.
- $\mathsf{Pair}(X, Y, N) \to \boldsymbol{E}$. 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.

1. 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.
2. For $p|N$, we have $\mathsf{Enc}i(X, N)_1 \bmod p = \mathsf{Enc}i(X, p)_1$, for $i = 1, 2$.

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

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 §5, 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 (see the full version).

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* if the following holds. For $N \in \mathbb{N}$, if $R_N(X, Y) = 0$, let $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, N)$, $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, N)$, 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: *selectively secure* and *co-selectively secure master-key hiding* ($\mathsf{SMH}, \mathsf{CMH}$) in a bilinear group generator $\mathcal{G}$. We first define the 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:

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

where st denotes the state information and the oracles $\mathcal{O}_1, \mathcal{O}_2$ in each state are defined below. The subscripts $\alpha, \boldsymbol{h}$ for each oracle are omitted for simplicity.

- **Selective Security (SMH).** $\mathcal{O}^1$ can be queried once while $\mathcal{O}^2$ can be queried polynomially many times.
  $\mathcal{O}^1_{\mathsf{SMH},b}(Y^\star)$: $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y^\star, p_2); \boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_{p_2}^{(w_2+1)}$; return $\boldsymbol{C} \leftarrow g_2^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}$.
  $\mathcal{O}^2_{\mathsf{SMH},b}(X)$: If $R_{p_2}(X, Y^\star) = 1$, then return $\perp$;
  else, $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X, p_2); \boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_{p_2}^{m_2}$; 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 (CMH).** Both $\mathcal{O}^1, \mathcal{O}^2$ can be queried once.
  $\mathcal{O}^1_{\mathsf{CMH},b}(X^\star)$: $(\boldsymbol{k}; m_2) \leftarrow \mathsf{Enc1}(X^\star, p_2); \boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_{p_2}^{m_2}$; return $\boldsymbol{K} \leftarrow \begin{cases} g_2^{\boldsymbol{k}(0,\boldsymbol{r},\boldsymbol{h})} & \text{if } b = 0 \\ g_2^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} & \text{if } b = 1 \end{cases}$
  $\mathcal{O}^2_{\mathsf{CMH},b}(Y)$: If $R_{p_2}(X^\star, Y) = 1$, then return $\perp$;
  else, $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, p_2); \boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_{p_2}^{(w_2+1)}$; return $\boldsymbol{C} \leftarrow g_2^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})}$.

We define the advantage of $\mathcal{A}$ in the game $\mathsf{G} \in \{\mathsf{SMH}, \mathsf{CMH}\}$ relative to $\mathcal{G}$ 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 PPT attackers $\mathcal{A}$. If both hold, we say that it is *doubly selectively master-key hiding*.

Remark 3. The terms corresponding to parameter $\boldsymbol{h}$ (in particular, $g_2^{\boldsymbol{h}}$) need *not* be given out to the adversary. Intuitively, this is since the security of encoding will be employed in the semi-functional space, and the parameter then corresponds to *semi-functional parameter* $\hat{\boldsymbol{h}}$, which needs not be sent (*cf.* §2).

4.3 Generic Construction for Functional Encryption from 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 $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}$, $Z_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Run $n \leftarrow \mathsf{Param}(\kappa)$. Pick $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_N^n$ and $\alpha \xleftarrow{\$} \mathbb{Z}_N$. The public key is $\mathsf{PK} = \big(g_1, e(g_1,g_1)^\alpha, g_1^{\boldsymbol{h}}, Z_3\big)$. The master secret key is $\mathsf{MSK} = \alpha$.
- $\mathsf{Encrypt}(Y, M, \mathsf{PK})$: Upon input $Y \in \mathbb{Y}_N$, run $(\boldsymbol{c}; w_2) \leftarrow \mathsf{Enc2}(Y, N)$. Pick $\boldsymbol{s} = (s, s_1, \ldots, s_{w_2}) \xleftarrow{\$} \mathbb{Z}_N^{w_2+1}$. Output the ciphertext as $\mathsf{CT} = (\boldsymbol{C}, C_0)$:

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

  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 SK as

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

- $\mathsf{Decrypt}(\mathsf{CT}, \mathsf{SK})$: 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 the 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 $\hat{\boldsymbol{h}} \xleftarrow{\$} \mathbb{Z}_N^n$.
- $\mathsf{SFEncrypt}(Y, M, \mathsf{PK}, g_2, \hat{\boldsymbol{h}})$: 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 $\mathsf{CT} = (\boldsymbol{C}, C_0)$ as

$$\boldsymbol{C} = g_1^{\boldsymbol{c}(\boldsymbol{s},\boldsymbol{h})} g_2^{\boldsymbol{c}(\hat{\boldsymbol{s}},\hat{\boldsymbol{h}})} \in \mathbb{G}^{w_1}, \qquad C_0 = (e(g_1,g_1)^\alpha)^s M \in \mathbb{G}_T.$$

- $\mathsf{SFKeyGen}(X, \mathsf{MSK}, \mathsf{PK}, g_2, \mathsf{type}, \hat{\alpha}, \hat{\boldsymbol{h}})$: Upon inputs $X, \mathsf{MSK}, \mathsf{PK}, g_2$, and $\mathsf{type} \in \{1,2,3\}$, $\hat{\alpha} \in \mathbb{Z}_N$, 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 SK as

$$\boldsymbol{K} = \begin{cases} g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \cdot \boldsymbol{R}_3 & \text{if } \mathsf{type} = 1 \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})} \cdot \boldsymbol{R}_3 & \text{if } \mathsf{type} = 2 \\ g_1^{\boldsymbol{k}(\alpha,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{\alpha},\boldsymbol{0},\boldsymbol{0})} \cdot \boldsymbol{R}_3 & \text{if } \mathsf{type} = 3 \end{cases}$$

  Note that the input $\hat{\alpha}$ (resp., $\hat{\boldsymbol{h}}$) is not needed for type 1 (resp., type 3).

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

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

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

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

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

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

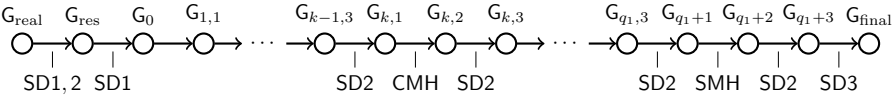
G_{q_1+3}:Modify $^{(4)}$ $\boxed{\mathsf{SK}_j \leftarrow \mathsf{SFKeyGen}(X_j, \mathsf{MSK}, \mathsf{PK}, g_2, 3, \hat{\alpha}, \mathbf{0})}$

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

Fig. 1. The sequence of games in the security proof

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*, we have* $\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) \le 2\mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}_1}(\lambda) + (2q_1+3)\mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{SD3}}_{\mathcal{B}_3}(\lambda) + q_1\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_4}(\lambda) + \mathsf{Adv}^{\mathsf{SMH}}_{\mathcal{B}_5}(\lambda)$*, where* q_1 *is the number of queries in phase 1.*

Security Proof. We use a sequence of games in the following order:



where each game is defined as follows. $\mathsf{G}_{\mathrm{real}}$ 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 are in the full version.

We also obtain the theorem for the case where the encoding is perfectly secure.

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* $\mathsf{FE}(\mathsf{P})$ *is fully secure. Indeed, let* $q_{\mathrm{all}} = q_1 + q_2$ *be the number of all queries. 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*,* $\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}_1}(\lambda) + (2q_{\mathrm{all}}+1)\mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{SD3}}_{\mathcal{B}_3}(\lambda)$.

5 Instantiations

5.1 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 [20]), 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 [20]). The novelty is that we can prove the SMH security (with tight reduction to $\mathsf{3DH}$). 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 $\mathsf{SD2}$ (plus tight reduction to $\mathsf{3DH}$, $\mathsf{SD1}$, $\mathsf{SD3}$). [5]

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

Theorem 3. *Scheme 1 is selectively master-key hiding under 3DH.*

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 $\mathsf{3DH}$. $\mathcal{B}$ takes as an input the $\mathsf{3DH}$ 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: $C_1 = g_2^s, C_2 = g_2^{s(h_1' + h_2' Y^\star)}, C_3 = (g_2^{h_3})^s$. Obviously, C_1, C_3 are properly distributed. C_2 is properly distributed due to the cancellation of unknown za in the exponent: $h_1 + h_2 Y^\star = (h_1' - Y^\star za) + (h_2' + za) Y^\star = h_1' + h_2' Y^\star$.

[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.

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 $K'_1=T((g_2^b)^{\frac{1}{X_j-Y^\star}})^{h'_1+h'_2X_j}$, $K'_2=(g_2^b)^{\frac{1}{X_j-Y^\star}}$, and $K'_3=1$. 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 $K'_1=g_2{}^{(\tau+abz)+(\frac{b}{X_j-Y^\star})(h'_1+h'_2X_j)}=g_2{}^{\tau+(\frac{b}{X_j-Y^\star})((h'_1-Y^\star za)+(h'_2+za)X_j)}=g_2{}^{\tau+r'_j(h_1+h_2X_j)}$, where the unknown element abz in the exponent term $r'_j(h_1+h_2X_j)$ is simulated by using abz from T. A crucial point here is that $\boldsymbol{K}'$ is not properly distributed yet 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 $K_1=K'_1(g_2{}^{r''_j})^{h'_1+h'_2X_j}(g_2^z)^{u''_j}$, $K_2=K'_2g_2{}^{r''_j}$, and $K_3=K'_3g_2{}^{u''_j}(g_2^a)^{-r''_j(X_j-Y^\star)}$. This is a properly distributed $\boldsymbol{K}=g_2{}^{\boldsymbol{k}_{X_j}(\tau,\boldsymbol{r}_j,\boldsymbol{h})}$ with $\boldsymbol{r}_j=(r_j,u_j)=(\frac{b}{X_j-Y^\star}+r''_j,u''_j-ar''_j(X_j-Y^\star))$.

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 [23].

5.2 Fully Secure FE for Regular Languages

Waters [34] proposed a selective secure FE scheme for regular languages. No fully secure realization has been known so far. [6] Our scheme is built upon [34].

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 $X,Y\in\{0,1\}^\ell$. To construct DFA that operates over small-size universe to determine if $X=Y$ would require $\Theta(\log\ell)$ transition, which might not be so satisfactory for such a simple primitive.

Our Fully Secure FE for Regular Languages. We propose a new scheme which is *fully secure* and operates over *large-universe* alphabet sets, *i.e.,* $|\Lambda|$ is

[6] Waters also suggested that dual system techniques could be used, but only with the restricted version of the primitive where some bounds must be posed. This is not satisfactory since the bound would negate the motivation of having arbitrary string sizes for the ciphertext attribute. A recent work [31] proposes such a bounded scheme.

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). Our encoding scheme is shown as Scheme 2.

Pair Encoding Scheme 2: FE for Regular Languages

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

$\mathsf{Enc1}(M) \to$ For any DFA $M = (Q, \mathbb{Z}_N, \mathcal{T}, q_0, q_{n-1})$, where $n = |Q|$, let $m = |\mathcal{T}|$, and parse $\mathcal{T} = \{(q_{x_t}, q_{y_t}, \sigma_t) | t \in [1, m]\}$.

$$\boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = \left(k_1, k_2, k_3, k_4, k_5, \{k_{6,t}, k_{7,t}, k_{8,t}\}_{t\in[1,m]}\right) :$$
$$\left\{ \begin{array}{lll} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_4 = r_0, & k_5 = -u_0 + r_0 h_0, & k_{6,t} = r_t, \\ k_{7,t} = u_{x_t} + r_t(h_1 + h_2\sigma_t), & k_{8,t} = -u_{y_t} + r_t(h_3 + h_4\sigma_t) & \end{array} \right\}$$

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

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

$$\boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = \left(c_1, c_2, c_3, c_4, \{c_{5,i}\}_{i\in[0,\ell]}, \{c_{6,i}\}_{i\in[1,\ell]}\right) :$$
$$\left\{ \begin{array}{lll} c_1 = s, & c_2 = s\eta, & c_3 = -s\phi_1 + s_\ell\phi_2, \\ c_4 = s_0 h_0, & c_{5,i} = s_i, & c_{6,i} = s_{i-1}(h_1 + h_2 w_i) + s_i(h_3 + h_4 w_i) \end{array} \right\}$$

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

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 $\rho_0, \rho_1, \cdots, \rho_n \in Q$ 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: $k_1 c_1 - k_2 c_2 + k_3 c_3 - k_4 c_4 + k_5 c_{5,0} + \sum_{i\in[1,\ell]}(-k_{6,t_i} c_{6,i} + k_{7,t_i} c_{5,i-1} + k_{8,t_i} c_{5,i}) = \alpha s$. 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.

We prove that Scheme 2 does not satisfy the perfectly master-key hiding security, by using some basic properties of DFA (see the full version). We then prove its SMH security under a new static assumption, EDHE1 (see below), which is similar to the assumption for Waters' scheme [34]. A notable difference is that the target element will be in $\mathbb{G}$ instead of $\mathbb{G}_T$ (similar to [23]). This is analogous to our IBE, where we use 3DH. The proof strategy for SMH of our encoding naturally follows from the selective security proof of Waters'. The harder part is to prove the CMH security (under another new static assumption), where we use completely new techniques. This is since there has been no known selectively secure FE for the *dual predicate* of regular languages functionality. One of our techniques is that we construct the scheme in such a way that both terms related to transitions in DFA (*i.e.,* $k_{7,t}, k_{8,t}$) are functions of the corresponding alphabet σ_t. This is in contrast with Waters' scheme where only one of them is a function of σ_t. The intuition is to perform a certain type of cancellation that comes from both terms, in the CMH proof. We state here only the assumption and the theorem for SMH, and postpone those for CMH to the full version.

Definition 3 (ℓ-EDHE1). *The ℓ-Expanded Diffie-Hellman Exponent Assumption-1 in subgroup $\mathbb{G}_{p_2}$ is defined as follows. Let $(\mathbb{G}, \mathbb{G}_T, e, N, p_1, p_2, p_3) \xleftarrow{\$} \mathcal{G}(\lambda)$ and $g_i \xleftarrow{\$} \mathbb{G}_{p_i}$. Let $a, b, c, d_1, \ldots, d_{\ell+1}, f, z \xleftarrow{\$} \mathbb{Z}_N$. Suppose that an adversary is given g_1, g_2, g_3, T, and $\boldsymbol{D}$ consisting of the following: $g_2^a, g_2^b, g_2^{a/f}, g_2^{1/f}, g_2^{a^\ell c/z}$, $\forall_{i\in[1,\ell+1]}\, g_2^{a^i/d_i}, g_2^{a^i bf}$; $\forall_{i\in[0,\ell]}\, g_2^{a^i c}, g_2^{bd_i}, g_2^{bd_i/f}, g_2^{abd_i/f}$; $\forall_{i\in[1,2\ell+1],i\neq\ell+1,\ j\in[1,\ell+1]}$ $g_2^{a^i c/d_j}$; $\forall_{i\in[2,2\ell+2],\ j\in[1,\ell+1]}\, g_2^{a^i bf/d_j}$; $\forall_{i,j\in[1,\ell+1],i\neq j}\, g_2^{a^i bd_j/d_i}$. Then, it is hard for any PPT adversary to distinguish whether $T = g_2^{abz}$ or $T \xleftarrow{\$} \mathbb{G}_{p_2}$.*

Theorem 4. *Scheme 2 is selectively master-key hiding under ℓ-EDHE1 with tight reduction, where ℓ is the length of the ciphertext query $w^\star$.*

5.3 Fully Secure ABE

Fully-Secure Unbounded ABE with Large Universes. Our pair encoding scheme for unbounded KP-ABE with large universes is shown as Scheme 3. 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 [26]. The correctness can be shown as follows. 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: $k_1c_1 - k_2c_2 - k_3c_3 + \sum_{i\in I} \mu_i \big(k_{4,i}c_4 - k_{5,i}c_{5,\pi(i)} + k_{6,i}c_{6,\pi(i)}\big) = \alpha s$.

Pair Encoding Scheme 3: Unbounded KP-ABE with Large Universes

$\mathsf{Param} \quad \to$ 6. Denote $\boldsymbol{h} = (h_0, h_1, \phi_1, \phi_2, \phi_3, \eta)$.

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

$$\left\{ \begin{array}{lll} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_{4,i} = A_i \boldsymbol{v}^\top + r_i\phi_3, & k_{5,i} = r_i, & k_{6,i} = r_i(h_0 + h_1\pi(i)) \end{array} \right\}$$

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

$\mathsf{Enc2}(S) \quad \to$ For $S \subseteq \mathbb{Z}_N$.
$\boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = \big(c_1, c_2, c_3, c_4, \{c_{5,y}, c_{6,y}\}_{y\in S}\big)$:

$$\left\{ \begin{array}{lll} c_1 = s, & c_2 = s\eta, & c_3 = s\phi_1 + w\phi_2, \\ c_4 = w, & c_{5,y} = w\phi_3 + s_y(h_0 + h_1 y), & c_{6,y} = s_y \end{array} \right\}$$

where $\boldsymbol{s} = (s, w, \{s_y\}_{y\in S})$.

Fully-Secure ABE with Short Ciphertexts. Our encoding for this primitive is shown as Scheme 4. Denote by T the maximum size for attribute sets S. 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]. The correctness can be shown as follows. When $R((A,\pi), S) = 1$, we have coefficients $\{\mu_i\}_{i\in I}$ similarly as above. Hence, we have $k_1c_1 - k_2c_2 - k_3c_3 + \sum_{i\in I} \mu_i \big(k_{4,i}c_4 - k_{5,i}c_5 + (\boldsymbol{k}_{6,i}(1,\boldsymbol{a})^\top)c_6\big) = \alpha s$, where $(1,\boldsymbol{a}) := (1, a_1, \ldots, a_T)$ and a_i is the coefficient of z^i in $p(z) = \prod_{y\in S}(z - y)$. Note that $\pi(i) \in S$ implies $p(\pi(i)) = 0$.

Pair Encoding Scheme 4: KP-ABE with Short Ciphertexts

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

$\mathsf{Enc1}(A, \pi) \to \boldsymbol{k}(\alpha, \boldsymbol{r}, \boldsymbol{h}) = \big(k_1, k_2, k_3, \{k_{4,i}, k_{5,i}, \boldsymbol{k}_{6,i}\}_{i\in[1,m]}\big)$:

For LSS $A \in \mathbb{Z}_N^{m\times k}$, $\pi : [1,m] \to \mathbb{Z}_N$ (π needed not be injective).

$$\left\{ \begin{array}{lll} k_1 = \alpha + r\phi_1 + u\eta, & k_2 = u, & k_3 = r, \\ k_{4,i} = A_i \boldsymbol{v}^\top + r_i\phi_3, & k_{5,i} = r_i, & \\ \boldsymbol{k}_{6,i} = \Big(r_i h_0, r_i\big(h_2 - h_1\pi(i)\big), \ldots, r_i\big(h_{T+1} - h_1\pi(i)^T\big)\Big) & & \end{array} \right\}$$

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

$\mathsf{Enc2}(S) \to \boldsymbol{c}(\boldsymbol{s}, \boldsymbol{h}) = \big(c_1, c_2, c_3, c_4, c_5, c_6\big)$:

For $S \subseteq \mathbb{Z}_N$ such that $|S| \le T$,
let a_i be the coefficient of z^i in $p(z) := \prod_{y\in S}(z-y)$.

$$\left\{ \begin{array}{llll} c_1 = s, & c_2 = s\eta, & c_3 = s\phi_1 + w\phi_2, & \\ c_4 = w, & c_5 = w\phi_3 + \tilde{s}(h_0 + h_1 a_0 + \cdots + h_{T+1}a_T), & c_6 = \tilde{s} & \end{array} \right\}$$

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

Both ABE schemes are special cases of our another new primitive called *key-policy over doubly spatial encryption*. We prove their SMH, CMH security under new static assumptions that are similar to those used for proving selective security of KP-ABE, CP-ABE of [26] respectively. Theses are provided in the full version. All the assumptions hold in the generic (bilinear) group model.

Acknowledgement. I would like to thank Michel Abdalla, Takahiro Matsuda, Shota Yamada, and reviewers for their helpful comments on previous versions.

References

1. Attrapadung, N., Libert, B.: Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 384–402. Springer, Heidelberg (2010)
2. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy 2007, pp. 321–334 (2007)
4. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity-Based encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
6. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)

7. Boneh, D., Sahai, A., Waters, B.: Functional Encryption: Definitions and Challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011)
8. Chen, J., Wee, H.: Fully (Almost) Tightly Secure IBE and Dual System Groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013)
9. Chen, C., Chen, J., Lim, H.W., Zhang, Z., Feng, D., Ling, S., Wang, H.: Fully Secure Attribute-Based Systems with Short Ciphertexts/Signatures and Threshold Access Structures. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 50–67. Springer, Heidelberg (2013)
10. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)
11. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013)
12. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits. In: FOCS 2013, pp. 40–49 (2013)
13. Goldwasser, S., Kalai, Y., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: STOC 2013, pp. 555–564 (2013)
14. Goldwasser, S., Kalai, Y., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to run Turing machines on encrypted data. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 536–553. Springer, Heidelberg (2013)
15. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: STOC 2013, pp. 545–554 (2013)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM CCS 2006, pp. 89–98 (2006)
17. Hamburg, M.: Spatial Encryption. Cryptology ePrint Archive: Report 2011/389
18. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
19. Lewko, A.: Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (2012)
20. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
21. Lewko, A., Waters, B.: Decentralizing Attribute-Based Encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011)
22. Lewko, A., Waters, B.: Unbounded HIBE and Attribute-Based Encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011)
23. Lewko, A., Waters, B.: New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012)

24. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
25. Naor, M., Reingold, O.: Number-Theoretic Constructions of Efficient Pseudo-Random Functions. Journal of ACM 51(2), 231–262 (2004)
26. Rouselakis, Y., Waters Practical, B.: constructions and new proof methods for large universe attribute-based encryption. In: ACM CCS 2013, pp. 463–474 (2013)
27. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
28. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
29. Okamoto, T., Takashima, K.: Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 591–608. Springer, Heidelberg (2012)
30. Okamoto, T., Takashima, K.: Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (2012)
31. Ramanna, S.C.: DFA-Based Functional Encryption: Adaptive Security from Dual System Encryption. Cryptology ePrint Archive: Report 2013/638
32. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
33. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
34. Waters, B.: Functional Encryption for Regular Languages. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 218–235. Springer, Heidelberg (2012)