Fully, (Almost) Tightly Secure IBE and Dual System Groups*

Jie Chen** and Hoeteck Wee***

[1] Nanyang Technological University, Singapore
[2] George Washington University

Abstract. We present the first fully secure Identity-Based Encryption scheme (IBE) from the standard assumptions where the security loss depends only on the security parameter and is independent of the number of secret key queries. This partially answers an open problem posed by Waters (Eurocrypt 2005). Our construction combines the Waters' dual system encryption methodology (Crypto 2009) with the Naor-Reingold pseudo-random function (J. ACM, 2004) in a novel way. The security of our scheme relies on the DLIN assumption in prime-order groups. Along the way, we introduce a novel notion of *dual system groups* and a new randomization and parameter-hiding technique for prime-order bilinear groups.

1 Introduction

In an Identity-Based Encryption (IBE) scheme [27], encryption requires only the identity of the recipient (e.g. an email address or an IP address) and a set of global public parameters, thus eliminating the need to distribute a separate public key for each user in the system. The first realizations of IBE were given in 2001; the security of these schemes were based on either Bilinear Diffie-Hellman or QR in the random oracle model [7, 13]. Since then, tremendous progress has been made towards obtaining IBE and HIBE schemes that are secure in the standard model based on pairings [8, 5, 6, 28, 15, 29] as well as lattices [16, 9, 2, 3]. Specifically, starting with [29], we now have very efficient constructions of IBE based on standard assumptions which achieve the strongest security notion of full (adaptive) security, where the adversary may choose the challenge identity after seeing both the public parameters and making key queries.

In this work, we focus on the issue of security reduction and security loss in the construction of fully secure IBE. Consider an IBE scheme with a security reduction showing that attacking the scheme in time t with success probability ϵ implies breaking some conjectured hard problem in time roughly t with success probability ϵ/L; we refer to L as the security loss, and a tight reduction is one where L is a constant. All

* This is a merge of two papers [10] and [11].

** Email: s080001@e.ntu.edu.sg. Supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03.

*** Email: hoeteck@alum.mit.edu. Supported by NSF CAREER Award CNS-1237429. Part of this work was done while visiting NTU.

known constructions of fully secure IBE schemes from standard assumptions incur a security loss that is at least linear in the number of key queries q; the only exceptions are constructions in the random oracle model [7] and those based on q-type assumptions [15]. Motivated by this phenomenon, Waters [28] posed the following problem in 2005 (reiterated in [15, 4]):

> “ *Design an IBE with a tight security reduction to a standard assumption.* ”

That is, we are interested in constructions based on “static” assumptions like the Decisional Linear (DLIN) assumption or the subgroup decisional assumption and which do not rely on random oracles. Note that an IBE with a tight security reduction would also imply signatures with a tight security reduction via the Naor's transformation [7]; indeed, the latter were the focus in a series of very recent works [1, 19, 17].

We stress that tight reductions are not just theoretical issues for IBE, rather they are of utmost practical importance: as L increases, we need to increase the size of the underlying groups in order to compensate for the security loss, which in turn increases the running time of the implementation. Note that the impact on performance is quite substantial, as exponentiation in a r-bit group takes time roughly $\mathcal{O}(r^3)$.

While the ultimate goal is to achieve constant security loss (i.e. $L = \mathcal{O}(1)$), even achieving $L = \mathrm{poly}(\lambda)$ and independent of q is already of both practical and theoretical interest. For typical settings of parameters (e.g. $\lambda = 128$ and $q = 2^{20}$), λ is much smaller than q. From the theoretical stand-point, we currently have two main techniques for obtaining fully secure IBE from standard assumptions: random partitioning [28] and dual system encryption framework [29]. For the former, we now know that an $\Omega(q)$ security loss is in fact inherent [18]. For the latter, all known instantiations also incur an $\Omega(q)$ security loss; an interesting theoretical question is whether this is in fact inherent to the dual system encryption framework.

1.1 Our results

Our main result is an IBE scheme based on the d-LIN assumption with security loss $\mathcal{O}(\lambda)$ for λ-bit identities:

Theorem 1. *There exists an IBE scheme for identity space $\{0,1\}^n$ based on the d-LIN assumption with the following property: for any adversary $\mathcal{A}$ that makes at most q key queries against the IBE scheme, there exist an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda) \leq (2n+1) \cdot \mathsf{Adv}^{d\text{-LIN}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$$

and

$$\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n),$$

where $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

We compare our scheme with prior constructions in Figure 1. Applying the Naor transform, we also obtain a d-LIN-based signature scheme with constant-size signatures and security loss independent of the number of signature queries. This yields an alternative construction for an analogous result in [17].

Reference	$\lvert\text{MPK}\rvert$	security loss	additive overhead	assumption
BB1 [5]	$\mathcal{O}(1)$	$\mathcal{O}(2^n)$	$q \cdot \mathrm{poly}(\lambda, n)$	DBDH
Waters [28]	$\mathcal{O}(n)$	$\mathcal{O}(qn)$	$q^2\epsilon^{-2} \cdot \mathrm{poly}(\lambda, n)$	DBDH
Gentry [15]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$q^2 \cdot \mathrm{poly}(\lambda, n)$	q-ABDHE
BR [4]	$\mathcal{O}(n)$	$\mathcal{O}(qn/\epsilon)$	$q \cdot \mathrm{poly}(\lambda, n)$	DBDH
LW[29, 22, 20]	$\mathcal{O}(1)$	$\mathcal{O}(q)$	$q \cdot \mathrm{poly}(\lambda, n)$	DLIN or composite
Ours	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$q \cdot \mathrm{poly}(\lambda, n)$	DLIN or composite
	$\mathcal{O}(d^2n)$	$\mathcal{O}(n)$	$d^2q \cdot \mathrm{poly}(\lambda, n)$	d-LIN

Fig. 1. Comparison amongst IBE schemes, where $\{0,1\}^n$ is the identity space, q is the number of adversary's key queries, and ϵ is the adversary's advantage. In all of these constructions, $\lvert\text{SK}\rvert = \lvert\text{CT}\rvert = \mathcal{O}(1)$.

Our approach. The inspiration for our construction comes from a recent connection between predicate encryption and one-time symmetric-key primitives [30] — namely one-time MACs in the case of IBE — via dual system encryption [29]. Our key observation is to extend this connection to "reusable MACs", namely that if we start with an appropriate pseudorandom function (PRF) with security loss L, we may derive an IBE with the security loss $\mathcal{O}(L)$. More concretely, we begin with the Naor-Reingold DDH-based PRF [24] which has security loss n for input domain $\{0,1\}^n$, and obtain a fully secure IBE with security loss $\mathcal{O}(n)$ via a novel variant of the dual system encryption methodology. Our IBE scheme is essentially that obtained by embedding Waters' fully secure IBE based on DBDH [28] into composite-order groups, and then converting this to a prime-order scheme following [10, 25, 20, 14] (along with some new technical ideas). Here, we exploit the fact that the Waters' IBE and the Naor-Reingold PRF share a similar algebraic structure based on bit-by-bit encoding of the identity and PRF input respectively.

1.2 Technical overview

We provide a more technical overview of our main results, starting with the proof idea and then the construction. Here, we assume some familiarity with prior works.

Proof idea. Our security proof combines Waters' dual system encryption methodology [29] with ideas from the analysis of the Naor-Reingold PRF. In a dual system encryption scheme [29], there are two types of keys and ciphertexts: normal and semi-functional. A key will decrypt a ciphertext properly unless both the key and the ciphertext are semi-functional, in which case decryption will fail with overwhelming probability. The normal keys and ciphertexts are used in the real system, and keys are gradually introduced in the hybrid security proof, one at a time. Ultimately, we arrive at a security game in which the simulator only has to produce semi-functional objects and security can be proved directly. In all prior instantiations of this methodology, the semi-functional keys are introduced one at a time. As a result, we require q hybrid games to switch all of the keys from normal to semi-functional, leading to an $\Omega(q)$ security loss, since each step requires a computational assumption.

We deviate from the prior paradigm by using only n hybrid games, iterating over the bits in the bit-by-bit encoding of the identity, as was done in the Naor-Reingold PRF. That is, we introduce n types of semi-functional ciphertexts and keys, where type i objects appear in game i, while gradually increasing the entropy in the semi-functional components in each game. This strategy introduces new challenges specific to the IBE setting, namely that the adversary could potentially use the challenge ciphertext to test whether we have switched from type $i-1$ keys to type i keys. Prior works exploit the fact that we only switch a *single* key in each step, whereas we could be switching up to q keys in each step.

We overcome this difficulty as follows. At step i of the hybrid game, we guess the i'th bit b_i of the challenge identity $\textsc{id}^*$, and abort if our guess is incorrect. This results in a security loss of 2, which we can afford. If our guess b_i is correct,

- for all identities whose i'th bit equals b_i, the corresponding type $i-1$ and type i object are the same;
- for all other identities, we increase the entropy of the keys going from type $i-1$ to type i (via a tight reduction to a computational assumption).

The first property implies that the adversary cannot use the challenge ciphertext to distinguish between type $i-1$ and type i keys; in the proof, the simulator will not be able to generate type $i-1$ or type i ciphertexts for identities whose i'th bit is different from b_i (c.f. Remark 3 and Section 4.4). Interestingly, decryption capabilities remain unchanged throughout the hybrid games: a type i key for $\textsc{id}^*$ can decrypt a type i ciphertext for $\textsc{id}^*$ (c.f. Remark 5). This is again different from prior instantiations of the dual system encryption methodology where decryption fails for semi-functional objects.

In the final transition, a semi-functional type n object for identity $\textsc{id}$ has semi-functional component $R_n(\textsc{id})$ where R_n is a truly random function. In particular, the semi-functional ciphertext has semi-functional component $R_n(\textsc{id}^*)$. Moreover, $R_n(\textsc{id}^*)$ is truly random from the adversary's view-point because it only learns $\textsc{sk}_{\textsc{id}}$ and thus $R_n(\textsc{id})$ for $\textsc{id} \neq \textsc{id}^*$. We can then argue that the message which is masked by $R_n(\textsc{id}^*)$ is information-theoretically hidden.

Construction. To achieve a modular analysis, we introduce a novel notion of nested dual system groups (see Section 3.1 for an overview). Our construction proceeds into two steps: the first builds an (almost) tight IBE from nested dual system groups where we rely on the Naor-Reingold PRF argument and the dual system encryption methodology; the second builds nested dual system groups from d-LIN where we handle all of the intricate linear algebra associated with simulating composite-order groups in prime-order groups from [10, 20] and with achieving a tight reduction via random self-reducibility.

Perspective. In spite of the practical motivation for tight security reductions, we clarify that our contributions are largely of theoretical and conceptual interest. This is because

Property	Where it is used	
	nested dual system groups	dual system groups
projective	correctness	correctness
	normal to type 0 (Lemma 1)	normal to semi-functional CT
associative	correctness	correctness
orthogonality	normal to type 0 (Lemma 2)	final transition
non-degeneracy	final transition (Lemma 4)	pseudo-normal to pseudo-SF keys
		final transition
$\mathbb{H}$-subgroup	type $i-1$ to type i (Lemma 3)	key delegation
left subgroup	normal to type 0 (Lemma 1)	normal to semi-functional CT
nested-hiding	type $i-1$ to type i (Lemma 3)	*unavailable*
right subgroup	*unavailable*	normal to pseudo-normal keys
		pseudo-SF to semi-functional keys
parameter-hiding	*unavailable*	pseudo-normal to pseudo-SF keys

Fig. 2. Summary of dual system groups (c.f. Section 3 and Appendix B)

any gain in efficiency from using smaller groups is overwhelmed by the loss from the bit-by-bit encoding of identities. Our work raises the following open problems:

- Can we reduce the size of the public parameters to a constant?
- Can we achieve tight security, namely $L = \mathcal{O}(1)$?

We note that progress on either problem would likely require improving on the Naor-Reingold PRF: namely, reducing respectively the seed length and the security loss to a constant, both of which are long-standing open problems. We also note that the present blow-up in public parameters and security loss arise only in using the Naor-Reingold approach to build an IBE from nested dual system groups; our instantiation of nested dual system groups do achieve tight security.

1.3 Additional results

As a pre-cursor to nested dual system groups, we introduce a basic notion of *dual system groups*. We present

- a generic construction of compact HIBE from dual system groups similar to the Lewko-Waters scheme over composite-order groups [22]; and
- instantiations of dual system groups under the d-LIN assumption in prime-order bilinear groups and the subgroup decisional assumption in composite-order bilinear groups respectively. Along the way, we provide a new randomization and parameter-hiding technique for prime-order groups.

Putting the two together, we obtain a new construction of compact HIBE in prime-order groups, as well as new insights into the structural properties needed for Waters' dual system encryption methodology [29]. We proceed to present an overview of dual system groups, our new techniques for prime-order groups and then an overview of nested dual system groups.

Dual system groups. Informally, dual system groups contain a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-generate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$. For concreteness, we may think of $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ as composite-order bilinear groups. Dual system groups take as input a parameter 1^n (think of n as the depth of the HIBE) and satisfy the following properties:

(subgroup indistinguishability.) There are two computationally indistinguishable ways to sample correlated $(n+1)$-tuples from $\mathbb{G}^{n+1}$: the "normal" distribution, and a higher-entropy distribution with "semi-functional components". An analogous statement holds for $\mathbb{H}^{n+1}$.

(associativity.) For all $(g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}$ and all $(h_0, h_1, \ldots, h_n) \in \mathbb{H}^{n+1}$ drawn from the respective normal distributions, we have that for all $i = 1, \ldots, n$,

$$e(g_0, h_i) = e(g_i, h_0).$$

(parameter-hiding.) Both normal distributions can be efficiently sampled given the public parameters; on the other hand, given only the public parameters, the higher-entropy distributions contain n "units" of information-theoretic entropy (in the semi-functional component), one unit for each of the n elements in the $(n+1)$-tuple apart from the first.

The key novelty in the framework lies in identifying the role of associativity in the prior instantiations of the dual system encryption methodology in composite-order groups [22].

Instantiation in prime-order groups. We present a new randomization and parameter-hiding technique for prime-order bilinear groups, which we use to instantiate dual system groups. This technique allows us to hide arbitrarily large amounts of entropy while working with a vector space of constant dimensions, whereas prior works require a linear blow-up in dimensions.

To motivate the new technique, we begin with a review of composite-order bilinear groups. Let (G_N, G_T) denote a composite-order bilinear group of order $N = p_1 p_2$ which is the product of two primes, endowed with an efficient bilinear map $e : G_N \times G_N \to G_T$. Let g denote an element of G_N of order p_1. A useful property of composite-order groups, especially in the context of dual system encryption [22, 23], is that we can perform randomization by raising a group element to the power of a random exponent $a \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. This operation satisfy the following useful properties:

(parameter-hiding.) given g, g^a, the quantity $a \pmod{p_2}$ is completely hidden;

(associativity.) for all $u \in G_N$, we have $e(g^a, u) = e(g, u^a)$.

We show how to achieve randomization in the prime-order setting under the d-LIN assumption. Fix a prime-order bilinear group (G, G_T) of order p, endowed with an efficient bilinear map $e : G \times G \to G_T$. Let g denote an element of G of order p. Elements in G_N correspond to elements in G^{d+1} and we consider the bilinear map $e : G^{d+1} \times G^{d+1} \to G_T$ given by $e(g^{\mathbf{x}}, g^{\mathbf{y}}) := e(g,g)^{\mathbf{x}^\top \mathbf{y}}$. Following [25, 14], we pick a random pair of orthogonal basis $(\mathbf{B}, \mathbf{B}^*) \leftarrow_{\mathrm{R}} \mathsf{GL}_{d+1}(\mathbb{Z}_p) \times \mathsf{GL}_{d+1}(\mathbb{Z}_p)$ so that $\mathbf{B}^\top \mathbf{B}^*$ is the identity matrix. We consider the projection maps π_L, π_R that map a $(d+1) \times (d+1)$ matrix to the left d columns and right-most column; they correspond to projecting $a \in \mathbb{Z}_N$ to $a \pmod{p_1}$ and $a \pmod{p_2}$ respectively.

We randomize a basis $(\mathbf{B}, \mathbf{B}^*)$ as follows: pick a random $\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}$ and replace $(\mathbf{B}, \mathbf{B}^*)$ with $(\mathbf{BA}, \mathbf{B}^*\mathbf{A}^\top)$. Observe that this transformation satisfy the following properties similar to those in the composite-order setting:

(parameter-hiding.) given $g^{\pi_L(\mathbf{B})}, g^{\pi_L(\mathbf{BA})}, g^{\pi_L(\mathbf{B}^*)}, g^{\pi_L(\mathbf{B}^*\mathbf{A}^\top)}$, the bottom-right entry of $\mathbf{A}$ is completely hidden;

(associativity.) for all $(\mathbf{B}, \mathbf{B}^*)$ and all $\mathbf{A} \in \mathbb{Z}_p^{(d+1)\times(d+1)}$, we have

$$e(g^{\mathbf{BA}}, g^{\mathbf{B}^*}) = e(g^{\mathbf{B}}, g^{\mathbf{B}^*\mathbf{A}^\top}) \ \left(= e(g,g)^{\mathbf{A}^\top}\right)$$

where $e(g^{\mathbf{X}}, g^{\mathbf{Y}}) := e(g,g)^{\mathbf{X}^\top \mathbf{Y}}$.

We also establish a subspace indistinguishability assumption similar to those in prior works [26, 20, 12].

Nested dual system groups. In nested dual system groups, we require a so-called *nested-hiding* property. Roughly speaking, this property says that it is computationally infeasible to distinguish q samples from some distribution with another; specifically, it allows us to boost the entropy of the semi-functional components. In the instantiation, we will need to establish this property with a tight reduction to some standard assumption. The nested-hiding property allows us to "embed" the Naor-Reingold analysis into the semi-functional space of a dual system encryption scheme. We stress that the nested-hiding property even for $q = 1$ is *qualitatively* different from right subgroup indistinguishability in dual system groups.

We outline the instantiations of dual system groups in the composite-order and prime-order settings:

- The composite-order instantiation is very similar to that as before. We rely on composite-order group whose order is the product of three primes p_1, p_2, p_3. The subgroup G_{p_1} of order p_1 serves as the "normal space" and G_{p_2} of order p_2 serves as the "semi-functional space". We also require a new static, generically secure assumption, which roughly speaking, states that DDH is hard in the G_{p_2} subgroup. Here, we extend the techniques from [24] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction. Our IBE analysis may also be viewed as instantiating the Naor-Reingold PRF in the G_{p_2} subgroup.

- For the prime-order instantiation based on d-LIN, we extend the prior instantiation in several ways. First, we work with $2d \times 2d$ matrices instead of $(d+1) \times (d+1)$ matrices. In both constructions, the first d dimensions serve as the "normal space"; in our construction, we require a d-dimensional semi-functional space instead of a 1-dimensional one so that we may embed the d-LIN assumption into the semi-functional space. Next, we extend the techniques from [24, 21] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction.

Perspective. In developing the framework for dual system groups, we opted to identify the minimal properties needed for the application to dual system encryption in the most basic setting of (H)IBE; we adopted an analogous approach also for nested dual system groups. An alternative approach would have been to maximize the properties satisfied by both the composite-order and prime-order instantiations, with the hope of capturing a larger range of applications. In choosing the minimalist approach, we believe we can gain better insights into how and why dual system encryption works, as well as guide potential lattice-based instantiations. In addition, we wanted the framework to be as concise as possible and the instantiations to be as simple as possible. Nonetheless, the framework remains fairly involved and we hope to see further simplifications in future work.

Organization. We present nested dual system groups in Section 3, our IBE scheme in Section 4 and a self-contained description of our d-LIN-based scheme in Appendix A. For completeness, we included a formal description of dual system group in Appendix B. We defer all other details to the full versions of this paper [11, 10].

2 Preliminaries

Notation. We denote by $s \leftarrow_{\mathrm{R}} S$ the fact that s is picked uniformly at random from a finite set S and by $x, y, z \leftarrow_{\mathrm{R}} S$ that all x, y, z are picked independently and uniformly at random from S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout, we use 1^λ as the security parameter. We use $\cdot$ to denote multiplication (or group operation) as well as component-wise multiplication. We use lower case boldface to denote (column) vectors over scalars or group elements and upper case boldface to denote vectors of group elements as well as matrices.

Identity-Based Encryption. An IBE scheme consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\lambda, 1^n) \rightarrow (\mathrm{MPK}, \mathrm{MSK})$. The setup algorithm takes in the security parameter 1^λ and the length parameter 1^n. It outputs public parameters MPK and a master secret key MSK.

$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m) \rightarrow \mathrm{CT}_{\mathbf{x}}$. The encryption algorithm takes in the public parameters MPK, an identity $\mathbf{x}$, and a message m. It outputs a ciphertext $\mathrm{CT}_{\mathbf{x}}$.

$\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{y}) \rightarrow \text{SK}_{\mathbf{y}}$. The key generation algorithm takes in the public parameters MPK, the master secret key MSK, and an identity $\mathbf{y}$. It outputs a secret key $\text{SK}_{\mathbf{y}}$.

$\mathsf{Dec}(\text{MPK}, \text{SK}_{\mathbf{y}}, \text{CT}_{\mathbf{x}}) \rightarrow m$. The decryption algorithm takes in the public parameters MPK, a secret key $\text{SK}_{\mathbf{y}}$ for an identity $\mathbf{y}$, and a ciphertext $\text{CT}_{\mathbf{x}}$ encrypted under an identity $\mathbf{x}$. It outputs a message m if $\mathbf{x} = \mathbf{y}$.

Correctness. For all $(\text{MPK}, \text{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, all identities $\mathbf{x}$, all messages m, all decryption keys $\text{SK}_{\mathbf{y}}$, all $\mathbf{x}$ such that $\mathbf{x} = \mathbf{y}$, we have

$$\Pr[\mathsf{Dec}(\text{MPK}, \text{SK}_{\mathbf{y}}, \mathsf{Enc}(\text{MPK}, \mathbf{x}, m)) = m] = 1.$$

Security Model. The security game is defined by the following experiment, played by a challenger and an adversary $\mathcal{A}$.

Setup. The challenger runs the setup algorithm to generate (MPK, MSK). It gives MPK to the adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ adaptively requests keys for any identity $\mathbf{y}$ of its choice. The challenger responds with the corresponding secret key $\text{SK}_{\mathbf{y}}$, which it generates by running $\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{y})$.

Challenge. The adversary $\mathcal{A}$ submits two messages m_0 and m_1 of equal length and a challenge identity $\mathbf{x}^*$ with the restriction that $\mathbf{x}^*$ is not equal to any identity requested in the previous phase. The challenger picks $\beta \leftarrow_{\text{R}} \{0,1\}$, and encrypts m_β under $\mathbf{x}^*$ by running the encryption algorithm. It sends the ciphertext to the adversary $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue key queries for any identity $\mathbf{y}$ as in Phase 1 with the restriction that $\mathbf{y} \neq \mathbf{x}^*$.

Guess. The adversary $\mathcal{A}$ must output a guess β' for β.

The advantage $\mathsf{Adv}_{\mathcal{A}}^{\text{IBE}}(\lambda)$ of an adversary $\mathcal{A}$ is defined to be $|\Pr[\beta' = \beta] - 1/2|$.

Definition 1. *An IBE scheme is* fully secure *if all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\text{IBE}}(\lambda)$ is a negligible function in λ.*

3 Nested Dual System Groups

In this section, we present nested dual system groups, a variant of dual system groups with a notable difference: we require (computational) nested-hiding indistinguishability, in place of (computational) right subgroup indistinguishability and (information-theoretic) parameter-hiding. As noted in the introduction, the nested-hiding property even for $q = 1$ is *qualitatively* different from right subgroup indistinguishability in dual system groups.

3.1 Overview

Informally, nested dual system groups contain a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-generate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$. For concreteness, we may think of $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ as composite-order bilinear groups. Nested dual system groups take as input a parameter 1^n and satisfy the following properties:

(left subgroup $\mathbb{G}$.) There are two computationally indistinguishable ways to sample correlated $(n+1)$-tuples from $\mathbb{G}^{n+1}$: the "normal" distribution, and a higher-entropy distribution with "semi-functional components". We sample the normal distribution using SampG and the semi-functional components using $\widehat{\mathsf{SampG}}$.

(right subgroup $\mathbb{H}$.) There is a single algorithm SampH to sample correlated $(n+1)$-tuples from $\mathbb{H}^{n+1}$. We should think of these tuples as already having semi-functional components, generated by some distinguished element $h^* \in \mathbb{H}$. It is convenient to think of h^* as being orthogonal to each component in the normal distribution over $\mathbb{G}$ (c.f. orthogonality and Remark 1). On the other hand, we require that h^* is *not* orthogonal to the semi-functional components in $\mathbb{G}$ (c.f. non-degeneracy) in order to information-theoretically hide the message in the final transition.

(nested-hiding.) We require a computational assumption over $\mathbb{H}$ which we refer to as *nested-hiding*, namely that for each $i = 1, \dots, n$,

$$(h_0, h_i) \quad \text{and} \quad (h_0, h_i \cdot (h^*)^\gamma)$$

are computationally indistinguishable, where $(h_0, h_1, \dots, h_n)$ is sampled using SampH and γ is a random exponent. In the formal definition, we provide the adversary with q samples from these distributions, and in the instantiations, we provide a tight reduction (independent of q) to a static assumption such as DLIN.

(associativity.) For all $(g_0, g_1, \dots, g_n) \in \mathbb{G}^{n+1}$ and all $(h_0, h_1, \dots, h_n) \in \mathbb{H}^{n+1}$ sampled using SampG and SampH respectively, we have that for all $i = 1, \dots, n$,

$$e(g_0, h_i) = e(g_i, h_0).$$

We require this property for correctness.

3.2 Definitions

Syntax. Nested dual system groups consist of five randomized algorithms given by $(\mathsf{SampP}, \mathsf{SampGT}, \mathsf{SampG}, \mathsf{SampH})$ along with $\widehat{\mathsf{SampG}}$:

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, output public and secret parameters $(\textsc{pp}, \textsc{sp})$, where:

- $\textsc{pp}$ contains a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-generate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, a linear map μ defined on $\mathbb{H}$, along with some additional parameters used by $\mathsf{SampG}, \mathsf{SampH}$;

- given PP, we know $\mathrm{ord}(\mathbb{H})$ (i.e. the order of the group, which is independent of n) and can uniformly sample from $\mathbb{H}$;
- SP contains $h^* \in \mathbb{H}$ (where $h^* \neq 1$), along with some additional parameters used by $\widehat{\mathsf{SampG}}$;

$\mathsf{SampGT} : \mathrm{Im}(\mu) \to \mathbb{G}_{\mathrm{T}}$. (As a concrete example, suppose $\mu : \mathbb{H} \to \mathbb{G}_T$ and $\mathrm{Im}(\mu) = \mathbb{G}_{\mathrm{T}}$.)

$\mathsf{SampG}(\mathrm{PP})$: Output $\mathbf{g} \in \mathbb{G}^{n+1}$.

$\mathsf{SampH}(\mathrm{PP})$: Output $\mathbf{h} \in \mathbb{H}^{n+1}$.

$\widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})$: Output $\hat{\mathbf{g}} \in \mathbb{G}^{n+1}$.

The first four algorithms are used in the actual scheme, whereas the last algorithm is used only in the proof of security. We define SampG_0 to denote the first group element in the output of SampG, and we define $\widehat{\mathsf{SampG}}_0$ analogously.

Correctness. The requirements for correctness are as follows:

(projective.) For all $h \in \mathbb{H}$ and all coin tosses s, we have $\mathsf{SampGT}(\mu(h); s) = e(\mathsf{SampG}_0(\mathrm{PP}; s), h)$.

(associative.) For all

$$(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathrm{PP}),\ (h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathrm{PP}),$$

and for all $i = 1, \ldots, n$, we have $e(g_0, h_i) = e(g_i, h_0)$.

Security. The requirements for security are as follows (we defer a discussion to the end of this section):

(orthogonality.) $\mu(h^*) = 1$.

(non-degeneracy.) With probability $1 - 2^{-\Omega(\lambda)}$ over $\hat{g}_0 \leftarrow \widehat{\mathsf{SampG}}_0(\mathrm{PP}, \mathrm{SP})$, we have that $e(\hat{g}_0, h^*)^\alpha$ is identically distributed to the uniform distribution over $\mathbb{G}_T$, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$.

($\mathbb{H}$-subgroup.) The output distribution of $\mathsf{SampH}(\mathrm{PP})$ is the uniform distribution over a subgroup of $\mathbb{H}^{n+1}$.

(left subgroup indistinguishability.) For any adversary $\mathcal{A}$, we define the advantage function:

$$\mathsf{Adv}^{\mathsf{LS}}_{\mathcal{A}}(\lambda) := \left|\Pr[\ \mathcal{A}(\mathrm{PP}, \boxed{\mathbf{g}}) = 1\ ] - \Pr[\ \mathcal{A}(\mathrm{PP}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}) = 1\ ]\right|$$

where

$$(\textsc{pp},\textsc{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);$$
$$\mathbf{g} \leftarrow \mathsf{SampG}(\textsc{pp});\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp},\textsc{sp}).$$

For any $\mathbf{g} = (g_0,\ldots,g_n) \in \mathbb{G}^{n+1}$, and any $i \in [n]$, we use $\mathbf{g}_{-i}$ to denote $(g_0,\ldots,g_{i-1},g_{i+1},\ldots,g_n) \in \mathbb{G}^n$.

(nested-hiding indistinguishability.) For any adversary $\mathcal{A}$, we define the advantage function:

$$\begin{aligned}\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda,q) := \max_{i\in[n]} \big|&\Pr[\,\mathcal{A}(\textsc{pp},h^*,\hat{\mathbf{g}}_{-i},\boxed{\mathbf{h}^1,\ldots,\mathbf{h}^q}) = 1\,] \\ &- \Pr[\,\mathcal{A}(\textsc{pp},h^*,\hat{\mathbf{g}}_{-i},\boxed{\mathbf{h}'^1,\ldots,\mathbf{h}'^q}) = 1\,]\big|\end{aligned}$$

where

$$(\textsc{pp},\textsc{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);$$
$$\hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp},\textsc{sp});$$
$$\mathbf{h}^j := (h_{0,j},h_{1,j},\ldots,\boxed{h_{i,j}},\ldots,h_{n,j}) \leftarrow \mathsf{SampH}(\textsc{pp}),\ j=1,\ldots,q;$$
$$\mathbf{h}'^j := (h_{0,j},h_{1,j},\ldots,\boxed{h_{i,j}\cdot(h^*)^{\gamma_j}},\ldots,h_{n,j}),\ \gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})},\ j=1,\ldots,q.$$

Discussion. We provide additional justification and discussion on the preceding security properties.

Remark 1 (orthogonality). We may deduce from $\mu(h^*) = 1$ that $e(g_0,h^*) = 1$ for all $g_0 = \mathsf{SampG}_0(\textsc{pp};s)$: for all $\gamma \in \{0,1\}$,

$$\begin{aligned}e(g_0,(h^*)^\gamma) &= \mathsf{SampGT}(\mu((h^*)^\gamma);s) && \text{(by \textit{projective})}\\ &= \mathsf{SampGT}(\mu(h^*)^\gamma;s) && \text{(by linearity of } \mu)\\ &= \mathsf{SampGT}(1;s) && \text{(by \textit{orthogonality})}\end{aligned}$$

Thus, we have $e(g_0,h^*) = e(g_0,1) = 1$. For the instantiation from composite-order groups, h^* is orthogonal to each element in the output of SampG, that is,

$$e(g_0,h^*) = e(g_1,h^*) = \cdots = e(g_n,h^*) = 1$$

for all $(g_0,g_1,\ldots,g_n) \leftarrow \mathsf{SampG}(\textsc{pp})$. On the other hand, for the instantiation from prime-order groups, h^* is in general not orthogonal to $g_1,\ldots,g_n$.

Remark 2 ($\mathbb{H}$-subgroup). We rely on $\mathbb{H}$-subgroup to re-randomize the secret keys in the proof of security for queries that share the same i-bit prefix; see Section 4.4 case 3.

Remark 3 (indistinguishability). Observe that in left subgroup indistinguishability, the distinguisher does not get h^*; otherwise, it is possible to distinguish between the two distributions using orthogonality. It is also crucial that for nested-hiding, the distinguisher gets $\hat{\mathbf{g}}_{-i}$ and not $\hat{\mathbf{g}} := (\hat{g}_0,\hat{g}_1,\ldots,\hat{g}_n)$. (Looking ahead to the proof in Section 4.4, not having $\hat{\mathbf{g}}$ means that the simulator cannot generate ciphertexts to distinguish between Type $i-1$ and Type i secret keys.) Otherwise, given $\hat{g}_i$, it is possible to distinguish between $\mathbf{h}^j$ and $\mathbf{h}'^j$ by using the relation:

$$e(g_0 \cdot \hat{g}_0, h_{i,j}) = e(g_i \cdot \hat{g}_i, h_{0,j}).$$

This relation follows from associative and left subgroup indistinguishability.

4 (Almost) Tight IBE from Nested Dual System Groups

We provide a construction of an IBE scheme from nested dual system groups where the ciphertext comprises two group elements in $\mathbb{G}$ and one in $\mathbb{G}_T$.

Overview. We begin with an informal overview of the scheme. Fix a bilinear group with a pairing $e : G \times G \to G_T$. The starting point of our scheme is the following variant of Waters' IBE [28] with identity space $\{0,1\}^n$:

$$\begin{aligned} \text{MPK} &:= (g, u_1, \ldots, u_{2n}, e(g,g)^\alpha) \\ \text{CT}_{\mathbf{x}} &:= (g^s, (\prod_{k=1}^{n} u_{2k-x_k})^s, e(g,g)^{\alpha s} \cdot m) \\ \text{SK}_{\mathbf{y}} &:= (g^r, \text{MSK} \cdot (\prod_{k=1}^{n} u_{2k-y_k})^r) \end{aligned}$$

Note that MPK contains $2n+1$ group elements in G, which we will generate using $\mathsf{SampP}(1^\lambda, \boxed{1^{2n}})$. We will use $\mathsf{SampG}(\text{PP})$ to generate the terms $(g^s, u_1^s, \ldots, u_{2n}^s)$ in the ciphertext, and $\mathsf{SampH}(\text{PP})$ to generate the terms $(g^r, u_1^r, \ldots, u_{2n}^r)$ in the secret key.

4.1 Construction

Let $\{0,1\}^n$ be the identity space.

- $\mathsf{Setup}(1^\lambda, 1^n)$: On input length parameter 1^n, first sample

  $$(\text{PP}, \text{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^{2n}).$$

  Pick $\text{MSK} \leftarrow_{\text{R}} \mathbb{H}$ and output the master public and secret key pair

  $$\text{MPK} := (\ \text{PP},\ \mu(\text{MSK})\ ) \quad \text{and} \quad \text{MSK}.$$

- $\mathsf{Enc}(\text{MPK}, \mathbf{x}, m)$: On input an identity $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$ and $m \in \mathbb{G}_T$, sample

  $$(g_0, g_1, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(\text{PP}; s),\ g'_T \leftarrow \mathsf{SampGT}(\mu(\text{MSK}); s)$$

  and output

  $$\text{CT}_{\mathbf{x}} := (\ C_0 := g_0,\ C_1 := g_{2-x_1} \cdots g_{2n-x_n},\ C_2 := g'_T \cdot m\ ) \in (\mathbb{G})^2 \times \mathbb{G}_T.$$

– $\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{y})$: On input an identity $\mathbf{y} \in \{0,1\}^n$, sample

$$(h_0, h_1, \ldots, h_{2n}) \leftarrow \mathsf{SampH}(\text{PP})$$

and output

$$\text{SK}_{\mathbf{y}} := (\ K_0 := h_0,\ K_1 := \text{MSK} \cdot h_{2-y_1} \cdots h_{2n-y_n}\ ) \in (\mathbb{H})^2.$$

– $\mathsf{Dec}(\text{MPK}, \text{SK}_{\mathbf{y}}, \text{CT}_{\mathbf{x}})$: If $\mathbf{x} = \mathbf{y}$, compute

$$e(g_0, \text{MSK}) \leftarrow e(C_0, K_1)/e(C_1, K_0)$$

and recover the message as

$$m \leftarrow C_2 \cdot e(g_0, \text{MSK})^{-1} \in \mathbb{G}_T.$$

Correctness. Fix $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$, observe that

$$\begin{aligned} & e(C_0, K_1)/e(C_1, K_0) \\ & = e(g_0, \text{MSK} \cdot h_{2-x_1} \cdots h_{2n-x_n}) \cdot e(g_{2-x_1} \cdots g_{2n-x_n}, h_0)^{-1} \\ & = e(g_0, \text{MSK}) \cdot \Big(e(g_0, h_{2-x_1}) \cdots e(g_0, h_{2n-x_n})\Big) \cdot \Big(e(g_{2-x_1}, h_0) \cdots e(g_{2n-x_n}, h_0)\Big)^{-1} \\ & = e(g_0, \text{MSK}) \end{aligned}$$

where the last equality relies on *associative*, namely, $e(g_0, h_{2i-x_i}) = e(g_{2i-x_i}, h_0)$. In addition, by *projective*, we have $g_T' = e(g_0, \text{MSK})$. Correctness follows readily.

4.2 Proof of Security

We prove the following theorem:

Theorem 2. *Under the left subgroup and nested-hiding indistinguishability (described in Section 3), our IBE scheme in Section 4.1 is fully secure (in the sense of Definition 1). More precisely, for any adversary $\mathcal{A}$ that makes at most q key queries against the IBE scheme, there exist adversaries $\mathcal{B}_1, \mathcal{B}_2$ such that:*

$$\mathsf{Adv}^{\text{IBE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{LS}}_{\mathcal{B}_1}(\lambda) + 2n \cdot \mathsf{Adv}^{\text{NS}}_{\mathcal{B}_2}(\lambda, q) + 2^{-\Omega(\lambda)}$$

and

$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot \text{poly}(\lambda, n),$$

where $\text{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Remark 4. In our instantiations of nested dual system groups, the quantity $\mathsf{Adv}^{\text{NS}}_{\mathcal{B}_2}(\lambda, q)$ will be related to the advantage function corresponding to some static assumption, with a constant overhead independent of q. Putting the two together, this means that $\mathsf{Adv}^{\text{IBE}}_{\mathcal{A}}(\lambda)$ is independent of q, as stated in Theorem 1.

The proof follows via a series of games, summarized in Figure 3. To describe the games, we must first define semi-functional keys and ciphertexts. Following [10, 30], we first define two auxiliary algorithms, and define the semi-functional distributions via these auxiliary algorithms.

Auxiliary algorithms. We consider the following algorithms:

$\widehat{\mathsf{Enc}}(\textsc{pp}, \mathbf{x}, m; \textsc{msk}', \mathbf{t})$: On input $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$, $m \in \mathbb{G}_T$, $\textsc{msk}' \in \mathbb{H}$, and $\mathbf{t} := (T_0, T_1, \ldots, T_{2n}) \in \mathbb{G}^{2n+1}$, output

$$\textsc{ct}_{\mathbf{x}} := \left(T_0,\ \prod_{k=1}^{n} T_{2k-x_k},\ e(T_0, \textsc{msk}') \cdot m \right).$$

$\widehat{\mathsf{KeyGen}}(\textsc{pp}, \textsc{msk}', \mathbf{y}; \mathbf{t})$: On input $\textsc{msk}' \in \mathbb{H}$, $\mathbf{y} := (y_1, \ldots, y_n) \in \{0,1\}^n$, and $\mathbf{t} := (T_0, T_1, \ldots, T_{2n}) \in \mathbb{H}^{2n+1}$, output

$$\textsc{sk}_{\mathbf{y}} := \left(T_0,\ \textsc{msk}' \cdot \prod_{k=1}^{n} T_{2k-y_k} \right).$$

Auxiliary distributions. For $i = 0, 1, \ldots, n$, we pick a random function $R_i : \{0,1\}^i \to \langle h^* \rangle$ (we use $\{0,1\}^0$ to denote the singleton set containing just the empty string ε). More concretely, given $(\textsc{pp}, h^*)$, we sample the function R_i by first choosing a random function $R_i' : \{0,1\}^i \to \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$ (via lazy sampling), and define $R_i(x) := (h^*)^{R_i'(x)}$ for all $x \in \{0,1\}^i$.

Pseudo-normal ciphertext.

$$\widehat{\mathsf{Enc}}(\textsc{pp}, \mathbf{x}, m; \textsc{msk}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}),$$

where $\mathbf{g} \leftarrow \mathsf{SampG}(\textsc{pp})$ and $\boxed{\hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp})}$; we can also write this distribution more explicitly as

$$\left(g_0 \cdot \hat{g}_0,\ \prod_{k=1}^{n} (g_{2k-x_k} \cdot \hat{g}_{2k-x_k}),\ e(g_0 \cdot \hat{g}_0, \textsc{msk}) \cdot m \right),$$

where $(g_0, g_1, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(\textsc{pp})$ and $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp})$.

Semi-functional ciphertext type i (for $i = 0, 1, \ldots, n$).

$$\widehat{\mathsf{Enc}}(\textsc{pp}, \mathbf{x}, m; \boxed{\textsc{msk} \cdot R_i(\mathbf{x}|_i)}, \mathbf{g} \cdot \hat{\mathbf{g}}),$$

where $\mathbf{g} \leftarrow \mathsf{SampG}(\textsc{pp})$ and $\hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp})$ and $\mathbf{x}|_i$ denotes the i-bit prefix of $\mathbf{x}$; we can also write this distribution more explicitly as

$$\left(g_0 \cdot \hat{g}_0,\ \prod_{k=1}^{n} (g_{2k-x_k} \cdot \hat{g}_{2k-x_k}),\ e(g_0 \cdot \hat{g}_0, \textsc{msk} \cdot R_i(\mathbf{x}|_i)) \cdot m \right),$$

where $(g_0, g_1, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(\textsc{pp})$ and $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp})$.

Semi-functional secret key type i (for $i = 0, 1, \ldots, n$).

$$\widehat{\mathsf{KeyGen}}(\textsc{pp}, \boxed{\textsc{msk} \cdot R_i(\mathbf{y}|_i)}, \mathbf{y}; \mathbf{h}),$$

Game	Ciphertext $\mathrm{CT}_{\mathbf{x}^*}$	Secret Key $\mathrm{SK}_{\mathbf{y}}$
0	$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}^*, m_\beta)$ $(g_0, \prod g_{2k-x_k}, e(g_0, \mathrm{MSK}) \cdot m_\beta)$	$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$ $(h_0, \mathrm{MSK} \cdot \prod h_{2k-y_k})$
1	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}})$ $(g_0\hat{g}_0, \prod(g_{2k-x_k}\hat{g}_{2k-x_k}), e(g_0\hat{g}_0, \mathrm{MSK}) \cdot m_\beta)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h})$ (—, —)
2,i	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \boxed{\mathrm{MSK} \cdot R_i(\mathbf{x}^* \vert_i)}, \mathbf{g} \cdot \hat{\mathbf{g}})$ (—, —, $e(g_0\hat{g}_0, \mathrm{MSK} \cdot R_i(\mathbf{x}^* \vert_i)) \cdot m_\beta$)	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK} \cdot R_i(\mathbf{y} \vert_i)}, \mathbf{y}; \mathbf{h})$ (—, $\mathrm{MSK} \cdot R_i(\mathbf{y} \vert_i) \cdot \prod h_{2k-y_k}$)
3	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, \boxed{\mathtt{random}}; \mathrm{MSK} \cdot R_n(\mathbf{x}^*), \mathbf{g} \cdot \hat{\mathbf{g}})$ (—, —, $e(g_0\hat{g}_0, \mathrm{MSK} \cdot R_n(\mathbf{x}^*)) \cdot \mathtt{random}$)	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot R_n(\mathbf{y}), \mathbf{y}; \mathbf{h})$ (—, $\mathrm{MSK} \cdot R_n(\mathbf{y}) \cdot \prod h_{2k-y_k}$)

Fig. 3. Sequence of games, where we drew a box to highlight the differences between each game and the preceding one, a dash (—) means the same as in the previous game. Recall that $R_i : \{0,1\}^i \to \langle h^* \rangle$ is a random function. Here, the product Π denotes $\Pi_{k=1}^n$. We transition from Game_0 to Game_1 and from $\mathsf{Game}_{2,i-1}$ to $\mathsf{Game}_{2,i}$ using a computational argument via left subgroup and nested-hiding respectively; for the remaining transitions, we use a statistical argument via orthogonality and non-degeneracy.

where a fresh $\mathbf{h} \leftarrow \mathsf{SampH}(\mathrm{PP})$ is chosen for each secret key; we can also write this distribution more explicitly as

$$\Big(\ h_0,\ \mathrm{MSK} \cdot R_i(\mathbf{x}|_i) \cdot \prod_{k=1}^{n} h_{2k-y_k}\ \Big)$$

where $(h_0, h_1, \ldots, h_{2n}) \leftarrow \mathsf{SampH}(\mathrm{PP})$.

Remark 5 (decryption capabilities). As noted in the introduction, decryption capabilities remain the same through the hybrid games. Observe that a type i secret key for $\mathbf{x}^*$ can decrypt a type i ciphertext for $\mathbf{x}^*$ since they share $R_i(\mathbf{x}^*|_i)$. In addition, a type i secret key for $\mathbf{x}^*$ can decrypt a normal ciphertext for $\mathbf{x}^*$ because $e(g_0, R_i(\mathbf{x}^*|_i)) = 1$, which follows readily from $R_i(\mathbf{x}^*|_i) \in \langle h^* \rangle$ and $e(g_0, h^*) = 1$ (see Remark 1).

Game sequence. We present a series of games. We write $\mathsf{Adv}_{\mathsf{xx}}(\lambda)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathsf{xx}}$.

- Game_0: is the real security game (c.f. Section 2).
- Game_1: is the same as Game_0 except that the challenge ciphertext is pseudo-normal.
- $\mathsf{Game}_{2,i}$ for i from 0 to n, $\mathsf{Game}_{2,i}$ is the same as Game_1 except that the challenge ciphertext and all secret keys are of type i.
- Game_3: is the same as $\mathsf{Game}_{2,n}$, except that the challenge ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_T$.

In Game_3, the view of the adversary is statistically independent of the challenge bit β. Hence, $\mathsf{Adv}_3(\lambda) = 0$. We complete the proof by establishing the following sequence of lemmas.

4.3 Normal to Pseudo-Normal to Type 0

Lemma 1 (Game_0 **to** Game_1)**.** *For any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_1$ such that:*

$$|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \leq \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda),$$

and $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The adversary $\mathcal{B}_1$ gets as input

$$(\mathrm{PP}, \mathbf{t}),$$

where $\mathbf{t}$ is either $\mathbf{g}$ or $\mathbf{g} \cdot \hat{\mathbf{g}}$ and

$$\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP}),\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP}),$$

and proceeds as follows:

Setup. Pick $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$ and output

$$\mathrm{MPK} := (\ \mathrm{PP},\ \mu(\mathrm{MSK})\ ).$$

Key Queries. On input the j'th secret key query $\mathbf{y}$, output

$$\mathrm{SK}_{\mathbf{y}} \leftarrow \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})).$$

Ciphertext. Upon receiving a challenge identity $\mathbf{x}^*$ and two equal length messages m_0, m_1, pick $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and output

$$\mathrm{CT}_{\mathbf{x}^*} \leftarrow \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \mathbf{t}).$$

Guess. When $\mathcal{A}$ halts with output β', $\mathcal{B}_1$ outputs 1 if $\beta' = \beta$ and 0 otherwise.

Observe that when $\mathbf{t} = \mathbf{g}$, $\mathrm{CT}_{\mathbf{x}^*}$ is properly distributed as $\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}^*, m_\beta)$ from *projective*, the output is identical to that in Game_0; and when $\mathbf{t} = \mathbf{g} \cdot \hat{\mathbf{g}}$, the output is identical to that in Game_1. We may therefore conclude that: $|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \leq \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda)$. □

Lemma 2 (Game_1 **to** $\mathsf{Game}_{2,0}$)**.** *For any adversary $\mathcal{A}$,*

$$\mathsf{Adv}_1(\lambda) = \mathsf{Adv}_{2,0}(\lambda)$$

Proof. Observe that MSK and $\mathrm{MSK} \cdot R_0(\varepsilon)$ (where $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$) are identically distributed, so we may replace MSK in Game_1 by $\mathrm{MSK} \cdot R_0(\varepsilon)$. The resulting distribution is identically distributed to that in $\mathsf{Game}_{2,0}$ except we use $\mu(\mathrm{MSK} \cdot R_0(\varepsilon))$ instead of $\mu(\mathrm{MSK})$ in MPK. Now, by *orthogonality*, these two quantities are in fact equal. □

4.4 Type $i-1$ to Type i

We begin with an informal overview of our proof strategy. For simplicity, suppose the adversary only requests secret keys for two identities $\mathbf{y}_0$ and $\mathbf{y}_1$ that differ only in the i'th bit, that is,

$$\mathbf{y}_0 = (y_1, \ldots, y_{i-1}, \boxed{0}, y_{i+1}, \ldots, y_n) \quad \text{and} \quad \mathbf{y}_1 = (y_1, \ldots, y_{i-1}, \boxed{1}, y_{i+1}, \ldots, y_n)$$

Recall that Type $i-1$ secret keys for $\mathbf{y}_0$ and $\mathbf{y}_1$ are of the form:

$$\begin{aligned} \mathsf{SK}_{\mathbf{y}_0} &= \Big(h_0, \mathsf{MSK} \cdot \boxed{R_{i-1}(y_1, \ldots, y_{i-1})} \cdot h_{2-y_1} \cdots \boxed{h_{2i}} \cdots h_{2n-y_n} \Big) \quad \text{and} \\ \mathsf{SK}_{\mathbf{y}_1} &= \Big(h_0, \mathsf{MSK} \cdot \boxed{R_{i-1}(y_1, \ldots, y_{i-1})} \cdot h_{2-y_1} \cdots \boxed{h_{2i-1}} \cdots h_{2n-y_n} \Big) \end{aligned}$$

whereas Type i secret keys for $\mathbf{y}_0$ and $\mathbf{y}_1$ are of the form:

$$\begin{aligned} \mathsf{SK}_{\mathbf{y}_0} &= \Big(h_0, \mathsf{MSK} \cdot \boxed{R_i(y_1, \ldots, y_{i-1}, 0)} \cdot h_{2-y_1} \cdots \boxed{h_{2i}} \cdots h_{2n-y_n} \Big) \quad \text{and} \\ \mathsf{SK}_{\mathbf{y}_1} &= \Big(h_0, \mathsf{MSK} \cdot \boxed{R_i(y_1, \ldots, y_{i-1}, 1)} \cdot h_{2-y_1} \cdots \boxed{h_{2i-1}} \cdots h_{2n-y_n} \Big) \end{aligned}$$

In order to show that Type $i-1$ and Type i secret keys for $\mathbf{y}_0$ and $\mathbf{y}_1$ are indistinguishable, it suffices to show that

$$\begin{aligned} &(R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i}, R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i-1}) \quad \text{and} \\ &(R_i(y_1, \ldots, y_{i-1}, 0) \cdot h_{2i}, R_i(y_1, \ldots, y_{i-1}, 1) \cdot h_{2i-1}) \end{aligned}$$

are computationally indistinguishable (*).

Now, suppose for simplicity that the i'th bit of the identity $\mathbf{x}^*$ for challenge ciphertext is 1. Then, *nested-hiding indistinguishability* with index $2i$ tells us that

$$h_{2i} \quad \text{and} \quad h_{2i} \cdot (h^*)^\gamma$$

are computationally indistinguishable, where $\gamma \leftarrow_{\mathrm{R}} \mathbb{Z}_{|\mathbb{H}|}$. Moreover, this holds even if the distinguisher is given $\hat{\mathbf{g}}_{-2i}$, which we will need to simulate the semi-functional ciphertext for $\mathbf{x}^*$. (On the other hand, given only $\hat{\mathbf{g}}_{-2i}$, we cannot simulate semi-functional ciphertext for identities whose i'th bit is 0.) This means that

$$\begin{aligned} &(R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i}, R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i-1}) \quad \text{and} \\ &(R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i} \cdot (h^*)^\gamma, R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i-1}) \end{aligned}$$

are computationally indistinguishable, even given the semi-functional ciphertext for $\mathbf{x}^*$.

To achieve (*), we can then implicitly set:

$$\begin{aligned} R_i(y_1, \ldots, y_{i-1}, 0) &:= R_{i-1}(y_1, \ldots, y_{i-1}) \cdot (h^*)^\gamma \quad \text{and} \\ R_i(y_1, \ldots, y_{i-1}, 1) &:= R_{i-1}(y_1, \ldots, y_{i-1}) \end{aligned}$$

This corresponds to Case 2 and Case 1 below respectively.

More generally, we guess at random the i'th bit of $\mathbf{x}^*$ to be b_i and use nested-hiding indistinguishability with index $2i - \overline{b_i}$. In addition, we need to handle q keys and not just two keys, along with an additional complication arising from the fact that multiple queries may share the same i-bit prefix (see Case 3 below).

Lemma 3 ($\mathsf{Game}_{2,i-1}$ **to** $\mathsf{Game}_{2,i}$)**.** *For $i = 1, \ldots, n$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_2$ such that:*

$$|\mathsf{Adv}_{2,i-1}(\lambda) - \mathsf{Adv}_{2,i}(\lambda)| \leq 2\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q),$$

and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. On input $i \in [n]$, $\mathcal{B}_2$ picks a random bit $b_i \leftarrow_{\mathrm{R}} \{0,1\}$ (that is, it guesses the i'th bit of the challenge identity $\mathbf{x}^*$) and requests nested-hiding instantiation for index $2i - \overline{b_i}$. The adversary $\mathcal{B}_2$ gets as input

$$\left(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-(2i-\overline{b_i})}, \mathbf{t}_1, \ldots, \mathbf{t}_q\right),$$

where $(\mathbf{t}^1, \ldots, \mathbf{t}^q)$ is either $(\mathbf{h}^1, \ldots, \mathbf{h}^q)$ or $(\mathbf{h}'^1, \ldots, \mathbf{h}'^q)$ and

$$\mathbf{h}^j := (h_{0,j}, h_{1,j}, \ldots, h_{2n,j}) \leftarrow \mathsf{SampH}(\mathrm{PP}),$$
$$\mathbf{h}'^j := (h_{0,j}, h_{1,j}, \ldots, h_{2i-\overline{b_i},j} \cdot (h^*)^{\gamma_j}, \ldots, h_{2n,j}),$$

and proceeds as follows:

Setup. Pick $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$, and output

$$\mathrm{MPK} := (\ \mathrm{PP},\ \mu(\mathrm{MSK})\ ).$$

Programming R_{i-1}, R_i. Pick a random function $\tilde{R}_{i-1} : \{0,1\}^{i-1} \to \langle h^* \rangle$ (which we use to program R_{i-1}, R_i). Recall that we can sample a uniformly random element in $\langle h^* \rangle$ by raising h^* to a uniformly random exponent in $\mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$. For all prefixes $\mathbf{x}' \in \{0,1\}^{i-1}$, we implicitly set

$$R_i(\mathbf{x}' \| b_i) := \tilde{R}_{i-1}(\mathbf{x}') \quad \text{and} \quad R_{i-1}(\mathbf{x}') := \tilde{R}_{i-1}(\mathbf{x}').$$

(We set $R_i(\mathbf{x}' \| \overline{b_i})$ later.) This means that for any $\mathbf{x} = (x_1, \ldots, x_n)$ such that $x_i = b_i$, we have:

$$R_i(\mathbf{x}|_i) = R_{i-1}(\mathbf{x}|_{i-1}) = \tilde{R}_{i-1}(\mathbf{x}|_{i-1}).$$

Key Queries. On input the j'th secret key query $\mathbf{y} = (\mathbf{y}|_{i-1}, y_i, \ldots, y_n)$, we consider three cases:

- Case 1: $y_i = b_i$. Here, $\mathcal{B}_2$ can compute

  $$R_i(\mathbf{y}|_i) = R_{i-1}(\mathbf{y}|_{i-1}) = \tilde{R}_{i-1}(\mathbf{y}|_{i-1})$$

  and simply outputs

  $$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{y}|_{i-1}), \mathbf{y}; \tilde{\mathbf{h}}^j),$$

  where $\tilde{\mathbf{h}}^j \leftarrow \mathsf{SampH}(\mathrm{PP})$.

- Case 2: $y_i = \overline{b_i}$ and $R_i(\mathbf{y}|_i)$ has not been previously set. Here, we implicitly set

  $$R_i(\mathbf{y}|_{i-1} \| \overline{b_i}) := \tilde{R}_{i-1}(\mathbf{y}|_{i-1}) \cdot (h^*)^{\gamma_j},$$

where γ_j is as defined in the nested-hiding instantiation. Observe that this is the correct distribution since $R_i(\mathbf{y}|_{i-1}\|b_i)$ and $R_i(\mathbf{y}|_{i-1}\|\overline{b_i})$ are two independently random values. Then $\mathcal{B}_2$ outputs:

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{y}|_{i-1}), \mathbf{y}; \mathbf{t}^j).$$

- Case 3: $y_i = \overline{b_i}$ and $R_i(\mathbf{y}|_i)$ has been previously set. Let j' be the index of key query in which we set $R_i(\mathbf{y}|_i)$, recall that

  $$R_i(\mathbf{y}|_{i-1}\|\overline{b_i}) := \tilde{R}_{i-1}(\mathbf{y}|_{i-1}) \cdot (h^*)^{\gamma_{j'}}.$$

  Then $\mathcal{B}_2$ outputs:

  $$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{y}|_{i-1}), \mathbf{y}; \mathbf{t}^{j'} \cdot \tilde{\mathbf{h}}^j).$$

  where $\tilde{\mathbf{h}}^j \leftarrow \mathsf{SampH}(\mathrm{PP})$. Here, we rely on the $\mathbb{H}$-*subgroup* property to re-randomize $\mathbf{t}^{j'}$.

Ciphertext. Upon receiving a challenge identity $\mathbf{x}^* := (x_1^*, \ldots, x_n^*)$ and two equal length messages m_0, m_1 from $\mathcal{A}$, output a random bit and halt if $x_i^* \neq b_i$. Observe that up to the point when $\mathcal{A}$ submits $\mathbf{x}^*$, its view is statistically independent of b_i. Therefore, the probability that we halt is exactly $1/2$. Suppose that we do not halt, which means we have $x_i^* = b_i$. Hence, $\mathcal{B}_2$ knows

$$R_i(\mathbf{x}^*|_i) = R_{i-1}(\mathbf{x}^*|_{i-1}) = \tilde{R}_{i-1}(\mathbf{x}^*|_{i-1}).$$

Then, $\mathcal{B}_2$ picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and outputs the semi-functional challenge ciphertext as:

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{x}^*|_{i-1}), \mathbf{g} \cdot \hat{\mathbf{g}}),$$

Here, $\mathcal{B}_2$ picks $\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP})$, whereas $\mathbf{g}$ is as defined in the nested-hiding instantiation. Observe that $\mathcal{B}_2$ can compute the output of $\widehat{\mathsf{Enc}}$ using just $\hat{\mathbf{g}}_{-(2i-\overline{b_i})}$ since since $x_i^* = b_i$.

Guess. When $\mathcal{A}$ halts with output β', $\mathcal{B}_2$ outputs 1 if $\beta' = \beta$ and 0 otherwise.

Suppose $x_i^* = b_i$. Then, when $(\mathbf{t}^1, \ldots, \mathbf{t}^q) = (\mathbf{h}^1, \ldots, \mathbf{h}^q)$, the output is identical to that in $\mathsf{Game}_{2,i-1}$; and when $(\mathbf{t}^1, \ldots, \mathbf{t}^q) = (\mathbf{h}'^1, \ldots, \mathbf{h}'^q)$, the output is identical to that in $\mathsf{Game}_{2,i}$. Hence,

$$\begin{aligned}
&\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q) \\
&= \Big| \Pr[x_i^* \neq b_i] \cdot 0 + \Pr[x_i^* = b_i] \\
&\quad \cdot (\Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i-1}] - \Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i}]) \Big| \\
&= 1/2 \cdot \Big| \Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i-1}] - \Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i}] \Big| \\
&\geq 1/2 \cdot |\mathsf{Adv}_{2,i-1}(\lambda) - \mathsf{Adv}_{2,i}(\lambda)|.
\end{aligned}$$

We may therefore conclude that $|\mathsf{Adv}_{2,i-1}(\lambda) - \mathsf{Adv}_{2,i}(\lambda)| \leq 2\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q)$. □

4.5 Final Transition

Lemma 4 ($\mathsf{Game}_{2,n}$ **to** Game_3). *For any adversary $\mathcal{A}$:*

$$|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \leq 2^{-\Omega(\lambda)}.$$

Proof. Observe that the challenge ciphertext in $\mathsf{Game}_{2,n}$ is given by:

$$\widehat{\mathsf{Enc}}(\textsc{pp}, \mathbf{x}^*, m_\beta; \textsc{msk} \cdot R_n(\mathbf{x}^*), \mathbf{g} \cdot \hat{\mathbf{g}}) = (C_0, C_1, C_2' \cdot m_\beta),$$

where (C_0, C_1) depend only on $\mathbf{g} \cdot \hat{\mathbf{g}} = (g_0 \cdot \hat{g}_0, \ldots)$, and C_2' is given by:

$$C_2' = e(g_0 \cdot \hat{g}_0, \textsc{msk} \cdot R_n(\mathbf{x}^*)) = e(g_0 \cdot \hat{g}_0, \textsc{msk}) \cdot \boxed{e(\hat{g}_0, R_n(\mathbf{x}^*))},$$

where in the last equality, we use the fact that $e(g_0, R_n(\mathbf{x}^*)) = 1$ (see Remarks 1 and 5). In addition, $\textsc{mpk}$ and all of the secret key queries reveal no information about $R_n(\mathbf{x}^*)$. Then, by *non-degeneracy*, with probability $1 - 2^{-\Omega(\lambda)}$ over $\hat{g}_0$, we have $e(\hat{g}_0, R_n(\mathbf{x}^*))$ is uniformly distributed over $\mathbb{G}_T$. This implies that the challenge ciphertext is identically distributed to a semi-functional encryption of a random message in $\mathbb{G}_T$, as in Game_3. We may then conclude that: $|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \leq 2^{-\Omega(\lambda)}$. □

Remark 6. In our composite-order instantiation, we only have the weaker guarantee that $e(\hat{g}_0, R_n(\mathbf{x}^*))$ has at least 2λ bits of min-entropy, instead of being uniform over $\mathbb{G}_T$. We will modify the IBE scheme as follows: the message space is now $\{0,1\}^\lambda$, and we replace the term $g_T' \cdot m$ in the ciphertext with:

$$\mathsf{H}(g_T') \oplus m,$$

where $\mathsf{H} : \mathbb{G}_T \to \{0,1\}^\lambda$ is a pairwise independent hash function. By the left-over hash lemma, we still have $|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \leq 2^{-\Omega(\lambda)}$.

Acknowledgments. We thank Dennis Hofheinz and the anonymous reviewers for helpful feedback on the write-up.

References

[1] M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly-secure signatures from lossy identification schemes. In *EUROCRYPT*, pages 572–590, 2012.

[2] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[3] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, pages 98–115, 2010.

[4] M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In *EUROCRYPT*, pages 407–424, 2009.

[5] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[6] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.

[7] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[8] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[9] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[10] J. Chen and H. Wee. Dual systems groups and its applications — compact HIBE and more. IACR Cryptology ePrint Archive, 2013.

[11] J. Chen and H. Wee. Fully, (almost) tightly secure IBE from standard assumptions. IACR Cryptology ePrint Archive, 2013.

[12] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee. Shorter IBE and signatures via asymmetric pairings. In *Pairing*, pages 122–140, 2012.

[13] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[14] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, pages 44–61, 2010.

[15] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[16] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[17] D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In *CRYPTO*, pages 590–607, 2012. Also Cryptology ePrint Archive, Report 2012/311.

[18] D. Hofheinz, T. Jager, and E. Knapp. Waters signatures with optimal security reduction. In *Public Key Cryptography*, pages 66–83, 2012.

[19] S. A. Kakvi and E. Kiltz. Optimal security proofs for full domain hash, revisited. In *EUROCRYPT*, pages 537–553, 2012.

[20] A. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.

[21] A. B. Lewko and B. Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In *ACM Conference on Computer and Communications Security*, pages 112–120, 2009.

[22] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, pages 455–479, 2010.

[23] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.

[24] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[25] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.

[26] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010. Also, Cryptology ePrint Archive, Report 2010/563.

[27] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[28] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[29] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

[30] H. Wee. Dual systems encryption via predicate encodings. Manuscript, 2013.

A Concrete IBE scheme from d-LIN in prime-order groups

In this section, we show how the concrete IBE scheme from d-LIN works in prime-order bilinear groups (G_1, G_2, G_T, e). Recall that $\pi_L : \mathbb{Z}_p^{2d\times 2d} \to \mathbb{Z}_p^{2d\times d}$ is the projection map that maps a $2d \times 2d$ matrix to the left d columns.

$\mathsf{Setup}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, sample

$$\mathbf{B}, \mathbf{B}^*, \mathbf{R} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p),\ \mathbf{A}_1, \ldots, \mathbf{A}_{2n} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(2d)\times(2d)},\ \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d}$$

such that $\mathbf{B}^\top \mathbf{B}^* = \mathbf{I}$, and output the master public and secret key pair

$$\begin{aligned} \mathrm{MPK} &:= \Big(\, g_1^{\pi_L(\mathbf{B})}, g_1^{\pi_L(\mathbf{BA}_1)}, \ldots, g_1^{\pi_L(\mathbf{BA}_{2n})}; e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})} \,\Big) \\ &\quad \in (G_1^{2d\times d})^{2n+1} \times G_T^d, \\ \mathrm{MSK} &:= \Big(\, g_2^{\mathbf{k}}, g_2^{\mathbf{B}^*\mathbf{R}}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_{2n}^\top\mathbf{R}} \,\Big) \in G_2^{2d} \times (G_2^{2d\times 2d})^{2n+1}. \end{aligned}$$

$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m)$: On input an identity vector $\mathbf{x} := (x_1, \ldots, x_n) \in \mathbb{Z}_p^n$ and $m \in G_T$, pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and output

$$\mathrm{CT}_{\mathbf{x}} := \left(\begin{array}{l} C_0 := g_1^{\pi_L(\mathbf{B})\mathbf{s}},\ C_1 := g_1^{\pi_L(\mathbf{B}(\mathbf{A}_{2-x_1}+\cdots+\mathbf{A}_{2n-x_n}))\mathbf{s}} \\ C_2 := e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \cdot m \end{array}\right) \in (G_1^{2d})^2 \times G_T.$$

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$: On input an identity vector $\mathbf{y} := (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d}$ and output

$$\mathrm{SK}_{\mathbf{y}} := \Big(\, K_0 := g_2^{\mathbf{B}^*\mathbf{Rr}},\ K_1 := g_2^{\mathbf{k}+\mathbf{B}^*(\mathbf{A}_{2-y_1}+\cdots+\mathbf{A}_{2n-y_n})^\top\mathbf{Rr}} \,\Big) \in (G_2^{2d})^2.$$

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{x} = \mathbf{y}$, compute

$$e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \leftarrow e(C_0, K_1)/e(C_1, K_0),$$

and recover the message as

$$m \leftarrow C_2 \cdot e(g_1, g_2)^{-\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \in G_T.$$

B Dual System Groups

Syntax. Dual system groups consist of six randomized algorithms given by (SampP, SampGT, SampG, SampH) along with ($\widehat{\mathsf{SampG}}$, $\widehat{\mathsf{SampH}}$):

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, output public and secret parameters $(\textsc{pp}, \textsc{sp})$, where:

- $\textsc{pp}$ contains a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-generate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, a linear map μ defined on $\mathbb{H}$, along with some additional parameters used by $\mathsf{SampG}, \mathsf{SampH}$;
- given $\textsc{pp}$, we know $\mathrm{ord}(\mathbb{H})$ (i.e. the order of the group, which is independent of n) and can uniformly sample from $\mathbb{H}$;
- $\textsc{sp}$ contains $h^* \in \mathbb{H}$ (where $h^* \neq 1$), along with some additional parameters used by $\widehat{\mathsf{SampG}}$;

$\mathsf{SampGT} : \mathrm{Im}(\mu) \to \mathbb{G}_\mathrm{T}$.

$\mathsf{SampG}(\textsc{pp})$: Output $\mathbf{g} \in \mathbb{G}^{n+1}$.

$\mathsf{SampH}(\textsc{pp})$: Output $\mathbf{h} \in \mathbb{H}^{n+1}$.

$\widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp})$: Output $\hat{\mathbf{g}} \in \mathbb{G}^{n+1}$.

$\widehat{\mathsf{SampH}}(\textsc{pp}, \textsc{sp})$: Output $\hat{\mathbf{h}} \in \mathbb{H}^{n+1}$.

The first four algorithms are used in the actual scheme, whereas the last two algorithms are used only in the proof of security. We define SampG_0 to denote the first group element in the output of SampG, and we define $\widehat{\mathsf{SampG}}_0, \widehat{\mathsf{SampH}}_0$ analogously.

Correctness. The requirements for correctness are as follows:

(projective.) For all $h \in \mathbb{H}$ and all coin tosses s, we have $\mathsf{SampGT}(\mu(h); s) = e(\mathsf{SampG}_0(\textsc{pp}; s), h)$.

(associative.) For all $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\textsc{pp})$ and $(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\textsc{pp})$ and for all $i = 1, \ldots, n$, we have $e(g_0, h_i) = e(g_i, h_0)$.

($\mathbb{H}$-subgroup.) The output distribution of $\mathsf{SampH}(\textsc{pp})$ is the uniform distribution over a subgroup of $\mathbb{H}^{n+1}$.

Security. The requirements for security are as follows:

(orthogonality.) $\mu(h^*) = 1$.

(non-degeneracy.) For all $\hat{h}_0 \leftarrow \widehat{\mathsf{SampH}}_0(\textsc{pp}, \textsc{sp})$, h^* lies in the group generated by $\hat{h}_0$. For all $\hat{g}_0 \leftarrow \widehat{\mathsf{SampG}}_0(\textsc{pp}, \textsc{sp})$, we have $e(\hat{g}_0, h^*)^\alpha$ is identically distributed to the uniform distribution over $\mathbb{G}_T$, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$.

(left subgroup indistinguishability.) For any adversary $\mathcal{A}$, we define the advantage function:

$$\mathsf{Adv}^{\mathrm{LS}}_{\mathcal{A}}(\lambda) := \left|\Pr[\,\mathcal{A}(\textsc{pp}, \boxed{\mathbf{g}}) = 1\,] - \Pr[\,\mathcal{A}(\textsc{pp}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}) = 1\,]\right|$$

where

$$\begin{aligned}&(\textsc{pp}, \textsc{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\ &\mathbf{g} \leftarrow \mathsf{SampG}(\textsc{pp});\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp}).\end{aligned}$$

(right subgroup indistinguishability.) For any adversary $\mathcal{A}$, we define the advantage function:

$$\mathsf{Adv}^{\mathrm{RS}}_{\mathcal{A}}(\lambda) := \big|\Pr[\ \mathcal{A}(\textsc{pp}, h^*, \mathbf{g}\cdot\hat{\mathbf{g}}, \boxed{\mathbf{h}}) = 1\ ] - \Pr[\ \mathcal{A}(\textsc{pp}, h^*, \mathbf{g}\cdot\hat{\mathbf{g}}, \boxed{\mathbf{h}\cdot\hat{\mathbf{h}}}) = 1\ ]\big|$$

where

$$\begin{aligned}&(\textsc{pp}, \textsc{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\ &\mathbf{g} \leftarrow \mathsf{SampG}(\textsc{pp});\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp});\\ &\mathbf{h} \leftarrow \mathsf{SampH}(\textsc{pp});\ \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\textsc{pp}, \textsc{sp}).\end{aligned}$$

(parameter-hiding.) The following distributions are identically distributed

$$\{\textsc{pp}, h^*, \boxed{\hat{\mathbf{g}}, \hat{\mathbf{h}}}\} \quad \text{and} \quad \{\textsc{pp}, h^*, \boxed{\hat{\mathbf{g}}\cdot\hat{\mathbf{g}}', \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}'}\}$$

where

$$\begin{aligned}&(\textsc{pp}, \textsc{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\ &\hat{\mathbf{g}} = (\hat{g}_0, \ldots) \leftarrow \widehat{\mathsf{SampG}}(\textsc{pp}, \textsc{sp});\\ &\hat{\mathbf{h}} = (\hat{h}_0, \ldots) \leftarrow \widehat{\mathsf{SampH}}(\textsc{pp}, \textsc{sp});\\ &\gamma_1, \ldots, \gamma_n \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})};\\ &\hat{\mathbf{g}}' := (1, \hat{g}_0^{\gamma_1}, \ldots, \hat{g}_0^{\gamma_n}) \in \mathbb{G}^{n+1};\\ &\hat{\mathbf{h}}' := (1, \hat{h}_0^{\gamma_1}, \ldots, \hat{h}_0^{\gamma_n}) \in \mathbb{H}^{n+1}.\end{aligned}$$