Fully, (Almost) Tightly Secure IBE from Standard Assumptions*

Jie Chen[1,**] and Hoeteck Wee[2,***]

[1] East China Normal University, China
[2] École Normale Supérieure, France

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 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.

Table of Contents

* A preliminary version of this work appeared as a merge with [12] at CRYPTO 2013 [11].

** Email: s080001@e.ntu.edu.sg. Supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03. Work done while at Nanyang Technological University (NTU) in Singapore.

*** Email: hoeteck@alum.mit.edu. Supported in part by NSF Awards CNS-1237429 and CNS-1319021. Most of this work was done at George Washington University and while visiting NTU.

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 [9, 5, 6, 28, 15, 29] as well as lattices [16, 10, 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 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 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 (generalized) d-LIN assumption with security loss $\mathcal{O}(\lambda)$ for λ-bit identities:

Reference	$\lvert\mathsf{MPK}\rvert$	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, 23, 21]	$\mathcal{O}(1)$	$\mathcal{O}(q)$	$q \cdot \mathrm{poly}(\lambda, n)$	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, ϵ is the adversary's advantage, and additive overhead refers to that in the simulator's running time in the security reduction. In all of these constructions, $|\mathsf{SK}| = |\mathsf{CT}| = \mathcal{O}(1)$.

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* $\mathsf{Time}(\mathcal{A})$.

This follows by combining our constructions in Section 4 and Section 6 (see also Theorem 2 and Lemma 8 and 10). We compare our scheme with prior constructions in Figure 1. Applying Naor's transformation, 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].

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 [25] 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 [12, 26, 21, 14] (along with some new technical ideas). Here, we exploit the fact that 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. (The formal presentation and analysis of our scheme is entirely self-contained.)

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

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 (nested) dual system groups

Construction. As noted earlier, our IBE scheme is essentially that obtained by embedding Waters' fully secure IBE based on DBDH into composite-order groups, and then converting this to a prime-order scheme (see Section 4 for an overview of the scheme). To achieve a modular analysis, we rely on a novel variant of the dual system group framework in [12], with a so-called *nested-hiding* property (see Section 3.1 for an overview). Roughly speaking, this 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.

Next, we provide new instantiations of such dual system groups in the composite-order and prime-order settings:

- The composite-order instantiation is very similar to that in [12, 23]. 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 [25] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction (c.f. Lemmas 6 and 7). 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 [12] 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 [25, 22] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction (c.f. Lemmas 9 and 10).

The modular approach allows us to decouple our main result into two steps: the first builds an 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 [12, 21] 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 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.

Organization. We present nested dual system groups in Section 3 and our IBE scheme in Section 4. We present instantiations of dual system groups in Sections 5 and 6. We present a self-contained description of our IBE scheme in Section 7.

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. Given a group G, we use $\mathrm{ord}(G)$ to denote the smallest positive integer c such that $g^c = 1$ for all $g \in G$.

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 $\mathrm{CT}_{\mathbf{x}}$.

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y}) \rightarrow \mathrm{SK}_{\mathbf{y}}$. 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 $\mathrm{SK}_{\mathbf{y}}$.

$\mathsf{Dec}(\text{MPK}, \text{SK}_{\mathbf{y}}, \text{CT}_{\mathbf{x}}) \to m$. 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 $\mathsf{Adv}^{\text{IBE}}_{\mathcal{A}}(\lambda)$ 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 $\mathcal{A}$, its advantage $\mathsf{Adv}^{\text{IBE}}_{\mathcal{A}}(\lambda)$ 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 introduced in [12] 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 $\widehat{\mathsf{SampG}}$. (This is exactly the same as in [12].)

(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 $i = 1, \ldots, n$,

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

are computationally indistinguishable, where $(h_0, h_1, \ldots, h_n)$ 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 $(g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}$ and all $(h_0, h_1, \ldots, h_n) \in \mathbb{H}^{n+1}$ sampled using SampG and SampH respectively, we have that for all $i = 1, \ldots, n$,

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

We require this property for correctness. (This is exactly the same as in [12].)

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 $(\mathrm{PP}, \mathrm{SP})$, where:

- 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 $\mathrm{ord}(\mathbb{H})$ 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 $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathrm{PP})$ and $(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathrm{PP})$ 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:

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

where

$$(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);$$
$$\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP});\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP}).$$

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:

$$\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda, q) := \max_{i \in [n]} \left|\Pr[\ \mathcal{A}(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-i}, \boxed{\mathbf{h}^1, \ldots, \mathbf{h}^q}) = 1\ ] - \Pr[\ \mathcal{A}(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-i}, \boxed{\mathbf{h}'^1, \ldots, \mathbf{h}'^q}) = 1\ ]\right|$$

where

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

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 $g_0 = \mathsf{SampG}_0(\mathrm{PP}; s)$: for all $\gamma \in \{0, 1\}$,

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

Thus, we have $e(g_0, h^*) = e(g_0, 1) = 1$. For the instantiation from composite-order groupsin Section 5, 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 $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathrm{PP})$. On the other hand, for the instantiation from prime-order groupsin Section 6, h^* is in general not orthogonal to $g_1, \ldots, g_n$. (This is the same as in [12]).

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. (In [12], the same property to used to re-randomize secret keys in HIBE key delegation.)

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 $\{0, 1\}^n$:

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

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

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

- $\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{y})$: 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) and the additional requirement that* $\mathrm{ord}(\mathbb{H})$ *is prime, 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:*

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

and

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

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

Remark 4. In our instantiations of nested dual system groups, the quantity $\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q)$ will be related to the advantage function corresponding to some static assumption, with a constant overhead independent of q (see Lemmas 7 and 10). Putting the two together, this means that $\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda)$ 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 [12, 30], we first define two auxiliary algorithms, and define the semi-functional distributions via these auxiliary algorithms.

Auxiliary algorithms. We consider the following algorithms:

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

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

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

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

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 (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.

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m; \mathrm{MSK}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}),$$

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

$$\Big(g_0 \cdot \hat{g}_0,\ \prod_{k=1}^{n} (g_{2k-x_k} \cdot \hat{g}_{2k-x_k}),\ e(g_0 \cdot \hat{g}_0, \mathrm{MSK}) \cdot m \Big),$$

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

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

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m; \boxed{\mathrm{MSK} \cdot R_i(\mathbf{x}|_i)}, \mathbf{g} \cdot \hat{\mathbf{g}}),$$

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

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

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

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

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK} \cdot R_i(\mathbf{y}|_i)}, \mathbf{y}; \mathbf{h}),$$

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 throughout the hybrid games. A type i secret key for $\mathbf{x}^*$ in $\mathsf{Game}_{2,i}$ 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 $\mathsf{Adv}_{\mathrm{xx}}(\lambda)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xx}}$.

- 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 $\mathbb{G}_T$.

Game	Ciphertext $\mathrm{CT}_{\mathbf{x}^*}$	Secret Key $\mathrm{SK}_{\mathbf{y}}$
0 : real game	$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}^*, m_\beta)$ $(g_0, \prod g_{2k-x_k}, e(g_0, \mathrm{MSK}) \cdot m_\beta)$	$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$ $(h_0, \mathrm{MSK} \cdot \prod h_{2k-y_k})$
1 : pseudo-normal CT via left subgroup	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}})$ $(g_0\hat{g}_0, \prod(g_{2k-x_k}\hat{g}_{2k-x_k}), e(g_0\hat{g}_0, \mathrm{MSK}) \cdot m_\beta)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h})$ $(—, —)$
2,i : (CT, SK) type i via nested-hiding	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \boxed{\mathrm{MSK} \cdot R_i(\mathbf{x}^*\|_i)}, \mathbf{g} \cdot \hat{\mathbf{g}})$ $(—, —, e(g_0\hat{g}_0, \mathrm{MSK} \cdot R_i(\mathbf{x}^*\|_i)) \cdot m_\beta)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK} \cdot R_i(\mathbf{y}\|_i)}, \mathbf{y}; \mathbf{h})$ $(—, \mathrm{MSK} \cdot R_i(\mathbf{y}\|_i) \cdot \prod h_{2k-y_k})$
3 : final game	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, \boxed{\mathtt{random}}; \mathrm{MSK} \cdot R_n(\mathbf{x}^*), \mathbf{g} \cdot \hat{\mathbf{g}})$ $(—, —, e(g_0\hat{g}_0, \mathrm{MSK} \cdot R_n(\mathbf{x}^*)) \cdot \mathtt{random})$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot R_n(\mathbf{y}), \mathbf{y}; \mathbf{h})$ $(—, \mathrm{MSK} \cdot R_n(\mathbf{y}) \cdot \prod h_{2k-y_k})$

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 $\Pi_{k=1}^n$.

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 $\mathbf{t} = \mathbf{g}$, $\text{CT}_{\mathbf{x}^*}$ is properly distributed as $\mathsf{Enc}(\text{MPK}, \mathbf{x}^*, m_\beta)$ 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: $|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \leq \mathsf{Adv}^{\text{LS}}_{\mathcal{B}_1}(\lambda)$. □

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 $\text{MSK} \cdot R_0(\varepsilon)$ (where $\text{MSK} \leftarrow_{\text{R}} \mathbb{H}$) are identically distributed, so we may replace MSK in Game_1 by $\text{MSK} \cdot R_0(\varepsilon)$. The resulting distribution is identically distributed to that in $\mathsf{Game}_{2,0}$ except we use $\mu(\text{MSK} \cdot R_0(\varepsilon))$ instead of $\mu(\text{MSK})$ 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:

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

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

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

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

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

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 $\gamma \leftarrow_{\text{R}} \mathbb{Z}_{\text{ord}(\mathbb{H})}$. 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

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

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

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

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

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:*

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

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,

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

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:

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

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:

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

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

5 Instantiations in Composite-Order Bilinear Groups

In this section, we present an instantiation of nested dual system groups in composite-order bilinear groups (introduced in [8] and used in [20, 23, 24]). The construction is very similar to that in [12, 23]. We require a new static, generally secure assumption in composite-order groups. In addition, we extend the techniques from [25] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction (c.f. Lemmas 6 and 7). Specifically, the differences from the prior composite-order instantiation in [12] are as follows:

- To establish nested-hiding indistinguishability, we rely on a new, generically secure assumption, which basically asserts that the DDH problem is hard in the G_{p_2}-subgroup (see Assumption 2); this replaces a subgroup decisional assumption used in the prior work for right subgroup indistinguishability.
- In the prior construction, SampG and SampH sample coin tosses from $\mathbb{Z}_N$ whereas $\widehat{\mathsf{SampG}}$ and $\widehat{\mathsf{SampH}}$ sample coin tosses from $\mathbb{Z}_N^*$. In the current construction, all of $\mathsf{SampG}, \mathsf{SampH}, \widehat{\mathsf{SampG}}$ sample coin tosses from $\mathbb{Z}_N$. (This is for simplicity as we no longer need to achieve parameter-hiding.)
- The output of SampH have (semi-functional) G_{p_2}-components, whereas in the prior construction, only the output of $\widehat{\mathsf{SampH}}$ (but not SampH) has (semi-functional) G_{p_2}-components.

5.1 Composite-Order Bilinear Groups

A generator $\mathcal{G}$ takes as input a security parameter λ and outputs a description (G_N, G_T, e), where N is product of distinct primes of $\Theta(\lambda)$ bits, G_N and G_T are cyclic groups of order N (specified using their respective generators), and $e : G_N \times G_N \to G_T$ is a non-degenerate bilinear map. We require that the group operations in G_N and G_T as well the bilinear map e are computable in deterministic polynomial time with respect to λ. We consider groups G whose orders are products of three distinct primes p_1, p_2, p_3 (that is, $N = p_1 p_2 p_3$). For every divisor n of N, we denote by G_n the subgroup of G_N of order n. We use g_1, g_2, g_3 to denote random generators of the subgroups $G_{p_1}, G_{p_2}, G_{p_3}$ of order p_1, p_2, and p_3 respectively.

Assumption 1 *For any adversary $\mathcal{A}$, we define the advantage function:*

$$\mathsf{Adv}^{\mathrm{DS1}}_{\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1] \right|$$

where

$$\begin{aligned}
&(N, G_N, G_T, g_1, g_2, g_3, e) \leftarrow \mathcal{G}(1^\lambda);\\
&h_{123} \leftarrow_{\mathrm{R}} G_N;\\
&D := ((N, G_N, G_T, e); g_1, g_3, h_{123});\\
&T_0 \leftarrow_{\mathrm{R}} G_{p_1},\ T_1 \leftarrow_{\mathrm{R}} G_{p_1 p_2}.
\end{aligned}$$

Assumption 2 *For any adversary $\mathcal{A}$, we define the advantage function:*

$$\mathsf{Adv}^{\mathrm{DS2}}_{\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1] \right|$$

where

$$\begin{aligned}
&(N, G_N, G_T, g_1, g_2, g_3, e) \leftarrow \mathcal{G}(1^\lambda);\\
&x, \tau \leftarrow_{\mathrm{R}} \mathbb{Z}_N,\ z \leftarrow_{\mathrm{R}} \mathbb{Z}^*_{p_2},\ X_3, Y_3 \leftarrow_{\mathrm{R}} G_{p_3};\\
&D := ((N, G_N, G_T, e); g_1, g_2, g_3, g_2^x X_3, g_2^\tau Y_3);\\
&T_0 = g_2^{x\tau},\ T_1 = g_2^{x\tau + z}.
\end{aligned}$$

5.2 Construction

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, do:

- run $(N, G_N, G_T, g_1, g_2, g_3, e) \leftarrow \mathcal{G}(1^\lambda)$, where $\mathcal{G}(1^\lambda)$ is a symmetric composite-order group generator;
- define $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) := (G_N, G_N, G_T, e)$;
- define $\mu : G_N \to G_T$ by $\mu(h) := e(g_1, h)$;
- sample $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}^n_N$, $h_{123} \leftarrow_{\mathrm{R}} G_N$, $h^* \leftarrow_{\mathrm{R}} G_{p_2 p_3}$ (we assume that h_{123} is a generator of G_N and h^* is a generator of $G_{p_2 p_3}$; these occur with overwhelming probability) and $\mathbf{R}_3 \leftarrow_{\mathrm{R}} G^n_{p_3}$ (using g_3);

Output

$$\mathrm{PP} := \left((N, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); g_1, g_1^{\mathbf{w}}, h_{123}, h_{123}^{\mathbf{w}} \cdot \mathbf{R}_3, g_3 \right) \quad \text{and} \quad \mathrm{SP} := \left(h^*, g_2, g_2^{\mathbf{w}} \right).$$

Note that $\mathrm{ord}(\mathbb{H}) = N$ and $\mathrm{ord}(h^*) = p_2 p_3$.

$\mathsf{SampGT}(g_T)$: Pick $s \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ and output $g_T^s \in G_T$.

$\mathsf{SampG}(\mathrm{PP})$: Pick $s \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ and output $(g_1^s, g_1^{s\mathbf{w}}) \in G_{p_1}^{n+1}$.

$\mathsf{SampH}(\mathrm{PP})$: Pick $r \leftarrow_{\mathrm{R}} \mathbb{Z}_N$, $\mathbf{X}_3 \leftarrow_{\mathrm{R}} G_{p_3}^n$ and output $(h_{123}^r, h_{123}^{r\mathbf{w}} \cdot \mathbf{X}_3) \in G_N^{n+1}$. Clearly, this satisfies the $\mathbb{H}$-subgroup property.

$\widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})$: Pick $\hat{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_N$ and output $(g_2^{\hat{s}}, g_2^{\hat{s}\mathbf{w}}) \in G_{p_2}^{n+1}$.

Correctness. We check correctness properties as follows:

(projective.) For all $h \in G_N$ and $s \in \mathbb{Z}_N$, we have

$$\mathsf{SampGT}(\mu(h); s) = \mathsf{SampGT}(e(g_1, h); s) = e(g_1, h)^s = e(g_1^s, h) = e(\mathsf{SampG}_0(\mathrm{PP}; s), h).$$

(associative.) We may write $\mathbf{w} := (w_1, \ldots, w_n)$, then for all

$$(g_1^s, g_1^{sw_1}, \ldots, g_1^{sw_n}) \leftarrow \mathsf{SampG}(\mathrm{PP}) \quad \text{and} \quad (h_{123}^r, h_{123}^{rw_1} \cdot X_{3,1}, \ldots, h_{123}^{rw_n} \cdot X_{3,n}) \leftarrow \mathsf{SampH}(\mathrm{PP})$$

and for all $i = 1, \ldots, n$, we have

$$e(g_1^s, h_{123}^{rw_i} \cdot X_{3,i}) = e(g_1, h_{123})^{srw_i} = e(g_1^{sw_i}, h_{123}^r).$$

Security. We check security properties as follows:

(orthogonality.) This follows readily from the fact that g_1 and h^* lie in orthogonal subgroups G_{p_1} and $G_{p_2p_3}$.

(non-degeneracy.) For all $g_2^{\hat{s}} \leftarrow \widehat{\mathsf{SampG}}_0(\mathrm{PP}, \mathrm{SP}; \hat{s})$, we have

$$e(g_2^{\hat{s}}, h^*) = e(g_2, h^*)^{\hat{s}} \neq 1 \text{ (i.e., } \mathrm{ord}(e(g_2^{\hat{s}}, h^*)) = p_2)$$

whenever $\hat{s} \neq 0 \pmod{p_2}$, which occurs with probability $1 - 1/p_2$; thus, $e(g_2^{\hat{s}}, h^*)^{\alpha}$ has at least $\log p_2$ bits of min-entropy with probability $1 - 2^{-\Omega(\lambda)}$, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. Here, we use the fact that h^* is a generator of $G_{p_2p_3}$ and $\hat{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_N$.

($\mathbb{H}$-subgroup.) This follows readily from the fact that $\mathbb{Z}_N$ is an additive group.

We establish left subgroup and nested-hiding indistinguishability in next three subsections, under computational assumptions in composite-order groups.

5.3 Left Subgroup Indistinguishability

We may rewrite the corresponding advantage function as:

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

where

$$(\text{PP}, \text{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);$$
$$\mathbf{g} := (g_1^s, g_1^{s\mathbf{w}}),\ s \leftarrow_{\text{R}} \mathbb{Z}_N;$$
$$\hat{\mathbf{g}} := (g_2^{\hat{s}}, g_2^{\hat{s}\mathbf{w}}),\ \hat{s} \leftarrow_{\text{R}} \mathbb{Z}_N.$$

Lemma 5 (DS1 to LS). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\text{LS}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{DS1}}_{\mathcal{B}}(\lambda) + 1/p_1 + 1/p_2 + /p_3.$$

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

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

$$((N, G_N, G_T, e); g_1, g_3, h_{123}, T)\,,$$

where either $T \leftarrow_{\text{R}} G_{p_1}$ or $T \leftarrow_{\text{R}} G_{p_1 p_2}$, and proceeds as follows:

Simulating PP. Pick $\mathbf{w} \leftarrow_{\text{R}} \mathbb{Z}_N^n$ and output

$$\text{PP} := (\ (N, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e);\ g_1, g_1^{\mathbf{w}}, h_{123}, h_{123}^{\mathbf{w}}, g_3\ )\,.$$

Observe that PP is properly distributed as long as h_{123} is a generator of G_N; this occurs with probability at least $1 - 1/p_1 - 1/p_2 - 1/p_3$.

Simulating the challenge. Output $(T, T^{\mathbf{w}})$.

Observe that when $T \leftarrow_{\text{R}} G_{p_1}$, the output is identical to $(\text{PP}, \mathbf{g})$; and when $T \leftarrow_{\text{R}} G_{p_1 p_2}$, the output is identical to $(\text{PP}, \mathbf{g} \cdot \hat{\mathbf{g}})$. The lemma then follows readily. □

5.4 Many-Tuple Lemma

We want to prove a many-tuple lemma which will be used in the proofs. The proof is essentially the same as that in [25, Lemma 3.2].

Lemma 6. *There exists an efficient algorithm that on input* 1^q *and*

$$\left(\ g_2, g_3, g_2^x X_3, g_2^\tau Y_3,\ g_2^{x\tau + z}\ \right),$$

we can generate q tuples of the form:

$$(g_2^{\hat{r}_j} \cdot X_{3,j}, T_j)$$

where

$$T_j = \begin{cases} g_2^{\hat{r}_j \tau} \cdot Y_{3,j} & \text{if } z = 0 \pmod{p_2} \\ g_2^{\hat{r}_j \tau} \cdot Y_{3,j} \cdot g_2^{\gamma_j} & \text{if } z \neq 0 \pmod{p_2} \end{cases}$$

and $\hat{r}_j, \gamma_j \leftarrow_{\text{R}} \mathbb{Z}_N$, $X_{3,j}, Y_{3,j} \leftarrow_{\text{R}} G_{p_3}$.

Proof. The algorithm proceeds as follows: for $j = 1, \ldots, q$, we pick $\tilde{r}_j, \tilde{\gamma}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N$, $\tilde{X}_{3,j}, \tilde{Y}_{3,j} \leftarrow_{\mathrm{R}} G_{p_3}$ and output

$$(g_2^{\hat{r}_j} \cdot X_{3,j}, T_j) := \left((g_2^x X_3)^{\tilde{\gamma}_j} \cdot g_2^{\tilde{r}_j} \cdot \tilde{X}_{3,j},\ T^{\tilde{\gamma}_j} \cdot (g_2^\tau Y_3)^{\tilde{r}_j} \cdot \tilde{Y}_{3,j} \right),$$

where we have implicitly set $\hat{r}_j := \tilde{\gamma}_j x + \tilde{r}_j$. It is easy to see that $\hat{r}_j$ is uniformly distributed over $\mathbb{Z}_N$. Observe that the exponent in the G_{p_2}-component of T_j is given by:

$$\tilde{\gamma}_j(x\tau + z) + \tilde{r}_j\tau = \hat{r}_j \cdot \tau + \tilde{\gamma}_j z,$$

it remains to analyze the distribution of the G_{p_2}-components of the q tuples:

- The case $z = 0$ is straight-forward;
- If $z \neq 0$, we implicitly set $\gamma_j := \tilde{\gamma}_j z$. Observe that $(\hat{r}_j, \gamma_j)$ are pairwise-independent since $z \neq 0$.

The lemma follows readily. □

5.5 Nested-Hiding Indistinguishability

We may rewrite the corresponding advantage function as:

$$\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda, q) := \max_{i \in [n]} \left| \Pr[\ \mathcal{A}(\mathrm{PP}, h^*, \mathbf{g}_{-i}, \boxed{\mathbf{h}^1, \ldots, \mathbf{h}^q}) = 1\ ] - \Pr[\ \mathcal{A}(\mathrm{PP}, h^*, \mathbf{g}_{-i}, \boxed{\mathbf{h}'^1, \ldots, \mathbf{h}'^q}) = 1\ ] \right|$$

where

$$\begin{aligned}
&(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\
&\hat{s}, r_j, \gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_N,\ \mathbf{X}_{3,j} \leftarrow_{\mathrm{R}} G_{p_3}^n,\ j = 1, \ldots, q;\\
&\mathbf{g}_{-i} := (g_2^{\hat{s}}, g_2^{\hat{s}\mathbf{w}})_{-i};\\
&\mathbf{h}^j := (h^{r_j}, \boxed{h^{r_j \mathbf{w}} \cdot \mathbf{X}_{3,j}}),\ j = 1, \ldots, q;\\
&\mathbf{h}'^j := (h^{r_j}, \boxed{h^{r_j \mathbf{w}} \cdot \mathbf{X}_{3,j} \cdot (h^*)^{\gamma_j \mathbf{e}_i}}),\ j = 1, \ldots, q.
\end{aligned}$$

Lemma 7 (DS2 to NS). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda, q) \leq \mathsf{Adv}^{\mathrm{DS2}}_{\mathcal{B}}(\lambda).$$

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

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

$$(\ (N, G_N, G_T, e); g_1, g_2, g_3, g_2^x X_3, g_2^\tau Y_3, T\ ),$$

where T is either $g_2^{x\tau}$ or $g_2^{x\tau + z}$, additional input $i \in [n]$, and proceeds as follows:

Generating q tuples. Run the algorithm in Lemma 6 on input 1^q and $(g_2, g_3, g_2^x X_3, g_2^\tau Y_3, T)$ to obtain

$$(g_2^{\hat{r}_j} \cdot X_{3,j}, T_j),\ j = 1, \ldots, q.$$

Programming $\mathbf{w}$. Pick $\tilde{\mathbf{w}} \leftarrow_R \mathbb{Z}_N^n$ and implicitly set

$$\mathbf{w} := \tilde{\mathbf{w}} \pmod{p_1 p_3}$$
$$\mathbf{w} := \tilde{\mathbf{w}} + \tau \mathbf{e}_i \pmod{p_2}$$

Simulating auxiliary input $\text{PP}, h^*, \hat{\mathbf{g}}_{-i}$. Pick $\tilde{r}, \tilde{r}' \leftarrow_R \mathbb{Z}_N^*$, $\hat{s} \leftarrow_R \mathbb{Z}_N$, $\mathbf{R}_3' \leftarrow_R G_{p_3}^n$, set $h_{123} := (g_1 g_2 g_3)^{\tilde{r}}$, and output

$$\text{PP} := \left((N, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); g_1, g_1^{\tilde{\mathbf{w}}}, h_{123}, h_{123}^{\tilde{\mathbf{w}}} \cdot (g_2^{\tau} Y_3)^{\tilde{r} \mathbf{e}_i} \cdot \mathbf{R}_3', g_3 \right), \; h^* := (g_2 g_3)^{\tilde{r}'}, \; \hat{\mathbf{g}}_{-i} := (g_2^{\hat{s}}, g_2^{\hat{s} \tilde{\mathbf{w}}})_{-i}.$$

Simulating the challenge. For $j = 1, \ldots, q$, $\mathcal{B}$ picks $\tilde{r}_j \leftarrow_R \mathbb{Z}_N, \tilde{\mathbf{X}}_{3,j} \leftarrow_R G_{p_3}^n$ and outputs as the j'th challenge:

$$\left(h_{123}^{\tilde{r}_j} \cdot g_2^{\hat{r}_j} \cdot X_{3,j}, \; (h_{123}^{\tilde{r}_j} \cdot g_2^{\hat{r}_j} \cdot X_{3,j})^{\tilde{\mathbf{w}}} \cdot ((g_2^{\tau} Y_3)^{\tilde{r}_j \cdot \tilde{r}} \cdot T_j)^{\mathbf{e}_i} \cdot \tilde{\mathbf{X}}_{3,j} \right),$$

where $\mathcal{B}$ has implicitly set $h_{123}^{r_j} := h_{123}^{\tilde{r}_j} \cdot g_2^{\hat{r}_j} \cdot X_{3,j}$.

Observe that:

- if $T = g_2^{x\tau}$, then $T_j = g_2^{\hat{r}_j \tau} \cdot Y_{3,j}$ and thus the j'th output challenge equals $\hat{\mathbf{h}}^j$;
- If $T = g_2^{x\tau + z}$, then $T_j = g_2^{\hat{r}_j \tau} \cdot Y_{3,j} \cdot g_2^{\gamma_j}$. Now, observe that

  $$(h^*, g_2^{\gamma_j} \cdot Y_{3,j}) \quad \text{and} \quad (h^*, (h^*)^{\gamma_j} \cdot Y_{3,j})$$

  are identically distributed for $\gamma_j \leftarrow_R \mathbb{Z}_N$ and $Y_{3,j} \leftarrow_R G_{p_3}$. Thus, the j'th output challenge has the same distribution as $\hat{\mathbf{h}}'^j$.

The claim then follows readily. □

6 Instantiations from d-LIN in Prime-Order Groups

We provide an instantiation of nested dual system groups from d-LIN in prime-order bilinear groups. We extend the instantiation in [12] 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 [25, 22] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction (c.f. Lemmas 9 and 10).

Combined with our IBE scheme in Section 4, we obtain an IBE based on d-LIN with the following parameters (omitting the G_2 terms MPK):

$$|\text{MPK}| = 2d^2(2n+1)|G_1| + d|G_T| \quad \text{and} \quad |\text{SK}| = 4d|G_2| \quad \text{and} \quad |\text{CT}| = 4d|G_1| + |G_T|$$

A self-contained description of our IBE scheme is given in Section 7.

6.1 Prime-Order Bilinear Groups

A generator $\mathcal{G}$ takes as input a security parameter λ and outputs a description $(p, G_1, G_2, G_T, g_1, g_2, e)$, where p is a prime of $\Theta(\lambda)$ bits; G_1, G_2 and G_T are cyclic groups of order p; g_1, g_2 are generators of G_1 and G_2 respectively; and $e : G_1 \times G_2 \to G_T$ is a non-degenerate bilinear map.

Assumption 3 (d-LIN: the d-linear assumption in G_1) *For any adversary $\mathcal{A}$, we define the advantage function:*

$$\mathsf{Adv}^{d\text{-LIN}}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T_0) - \Pr[\mathcal{A}(D, T_1)]|$$

where

$$\begin{aligned}
&(p, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda);\\
&s_1, \ldots, s_d \leftarrow_{\mathrm{R}} \mathbb{Z}_p;\ a_1, \ldots, a_d, s_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*;\\
&D := ((p, G_1, G_2, G_T, e); g_1, g_2, g_1^{a_1}, \ldots, g_1^{a_d}, g_1^{a_{d+1}}, g_1^{a_1 s_1}, \ldots, g_1^{a_d s_d});\\
&T_0 := g_1^{a_{d+1}(s_1+\cdots+s_d)},\ T_1 := g_1^{a_{d+1}(s_1+\cdots+s_d)+s_{d+1}}.
\end{aligned}$$

Remark 7. Typically, we sample $a_1, \ldots, a_d, s_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p$; this yields a $(d+1)/p$ negligible difference in the advantage.

Matrix-in-the-exponent. Given two vectors $\mathbf{x} = (x_1, \ldots, x_n)^\top, \mathbf{y} = (y_1, \ldots, y_n)^\top$ over scalars, we use $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote the standard dot product $\mathbf{x}^\top \mathbf{y}$. Given a group element g, we write $g^{\mathbf{x}}$ to denote $(g^{x_1}, \ldots, g^{x_n})^\top$; we define $g^{\mathbf{A}}$ where $\mathbf{A}$ is a matrix in an analogous way. Note that given a matrix of group elements $g^{\mathbf{A}}$, and a matrix $\mathbf{B}$ of "exponents", one can efficiently compute $g^{\mathbf{AB}}$; we will also denote this computation by $(g^{\mathbf{A}})^{\mathbf{B}}$. On the other hand, if $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are three groups endowed with an efficient bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, then given $g_1^{\mathbf{A}}, g_2^{\mathbf{B}}$ for $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$, one can efficiently compute $e(g_1, g_2)^{\mathbf{A}^\top \mathbf{B}}$ via $(e(g_1, g_2)^{\mathbf{A}^\top \mathbf{B}})_{ij} = \prod_k e(g_1^{\mathbf{A}_{k,i}}, g_2^{\mathbf{B}_{k,j}})$. We will use $e(g_1^{\mathbf{A}}, g_2^{\mathbf{B}}) = e(g_1, g_2)^{\mathbf{A}^\top \mathbf{B}}$ to denote this operation.

6.2 Construction

Let π_L, π_R be the projection maps that map a $2d \times 2d$ matrix to the left d columns and right d columns respectively.

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, do:

- run $(p, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda)$, where $\mathcal{G}(1^\lambda)$ is an asymmetric prime-order group generator;
- define $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) := (G_1^{2d}, G_2^{2d}, G_T, e)$;
- sample $\mathbf{B}, \mathbf{R} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p)$, along with $\mathbf{A}_1, \ldots, \mathbf{A}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d \times 2d}$ and set $\mathbf{B}^* := (\mathbf{B}^{-1})^\top$; define

$$\begin{aligned}
&\mathbf{D} := \pi_L(\mathbf{B}),\ \mathbf{D}_i := \pi_L(\mathbf{B}\mathbf{A}_i),\ \mathbf{F} := \pi_R(\mathbf{B}),\ \mathbf{F}_i := \pi_R(\mathbf{B}\mathbf{A}_i)\\
&\mathbf{D}^* := \mathbf{B}^*\mathbf{R}, \quad \mathbf{D}_i^* := \mathbf{B}^*\mathbf{A}_i^\top\mathbf{R},
\end{aligned};$$

- define $\mu : G_2^{2d} \to G_T^d$ by $\mu(g_2^{\mathbf{k}}) = e(g_1^{\mathbf{D}}, g_2^{\mathbf{k}})$ for all $\mathbf{k} \in \mathbb{Z}_p^{2d}$;
- set $h^* := g_2^{\mathbf{B}^* \mathbf{e}_{2d}}$;

Output

$$\text{PP} := \left((p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); \begin{matrix} g_1^{\mathbf{D}}, & g_1^{\mathbf{D}_1}, \ldots, g_1^{\mathbf{D}_n} \\ g_2^{\mathbf{D}^*}, & g_2^{\mathbf{D}_1^*}, \ldots, g_2^{\mathbf{D}_n^*} \end{matrix} \right) \quad \text{and} \quad \text{SP} := \left(g_2^{\mathbf{B}^* \mathbf{e}_{2d}}, g_1^{\mathbf{F}}, g_1^{\mathbf{F}_1}, \ldots, g_1^{\mathbf{F}_n} \right).$$

Note that $\mathrm{ord}(\mathbb{H}) = p$ and $\mathrm{ord}(h^*) = p$.

$\mathsf{SampGT}(g_T^{\mathbf{p}})$: Pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and output $g_T^{\mathbf{s}^\top \mathbf{p}} \in G_T$.

$\mathsf{SampG}(\text{PP})$: Pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and output $(g_1^{\mathbf{D}\mathbf{s}}, g_1^{\mathbf{D}_1 \mathbf{s}}, \ldots, g_1^{\mathbf{D}_n \mathbf{s}}) \in (G_1^{2d})^{n+1}$.

$\mathsf{SampH}(\text{PP})$: Pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d}$ and output $(g_2^{\mathbf{D}^* \mathbf{r}}, g_2^{\mathbf{D}_1^* \mathbf{r}}, \ldots, g_2^{\mathbf{D}_n^* \mathbf{r}}) \in (G_2^{2d})^{n+1}$. Clearly, this satisfies the $\mathbb{H}$-subgroup property.

$\widehat{\mathsf{SampG}}(\text{PP}, \text{SP})$: Pick $\hat{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and output $(g_1^{\mathbf{F}\hat{\mathbf{s}}}, g_1^{\mathbf{F}_1 \hat{\mathbf{s}}}, \ldots, g_1^{\mathbf{F}_n \hat{\mathbf{s}}}) \in (G_1^{2d})^{n+1}$.

Correctness. We check correctness properties as follows:

(projective.) For all $\mathbf{k} \in \mathbb{Z}_p^{2d}$ and all coin tosses $\mathbf{s} \in \mathbb{Z}_p^d$, we have $\mu(g_2^{\mathbf{k}}) = e(g_1^{\mathbf{D}}, g_2^{\mathbf{k}})$ and

$$\mathsf{SampGT}(\mu(g_2^{\mathbf{k}}); \mathbf{s}) = e(g_1, g_2)^{\mathbf{s}^\top (\mathbf{D}^\top \mathbf{k})} = e(g_1^{\mathbf{D}\mathbf{s}}, g_2^{\mathbf{k}}) = e(\mathsf{SampG}_0(\text{PP}; \mathbf{s}), g_2^{\mathbf{k}}),$$

where in the second equality, we use the fact that $\mathbf{s}^\top (\mathbf{D}^\top \mathbf{k}) = (\mathbf{D}\mathbf{s})^\top \mathbf{k}$.

(associative.) We need to show that for all

$$(g_1^{\mathbf{D}\mathbf{s}}, g_1^{\mathbf{D}_1 \mathbf{s}}, \ldots, g_1^{\mathbf{D}_n \mathbf{s}}) \leftarrow \mathsf{SampG}(\text{PP}) \quad \text{and} \quad (g_2^{\mathbf{D}^* \mathbf{r}}, g_2^{\mathbf{D}_1^* \mathbf{r}}, \ldots, g_2^{\mathbf{D}_n^* \mathbf{r}}) \leftarrow \mathsf{SampH}(\text{PP})$$

and for all $i = 1, \ldots, n$, we have

$$e(g_1^{\mathbf{D}\mathbf{s}}, g_2^{\mathbf{D}_i^* \mathbf{r}}) = e(g_1^{\mathbf{D}_i \mathbf{s}}, g_2^{\mathbf{D}^* \mathbf{r}}).$$

Observe that for all i,

$$\mathbf{B}^\top (\mathbf{B}^* \mathbf{A}_i^\top \mathbf{R}) = (\mathbf{B}^\top \mathbf{B}^*) \mathbf{A}_i^\top \mathbf{R} = \mathbf{A}_i^\top \mathbf{R} = \mathbf{A}_i^\top (\mathbf{B}^\top \mathbf{B}^*) \mathbf{R} = (\mathbf{B}\mathbf{A}_i)^\top (\mathbf{B}^* \mathbf{R}).$$

This implies

$$[\mathbf{D} \| \mathbf{F}]^\top \mathbf{D}_i^* = [\mathbf{D}_i \| \mathbf{F}_i]^\top \mathbf{D}^*$$

and thus $\mathbf{D}^\top \mathbf{D}_i^* = \mathbf{D}_i^\top \mathbf{D}^*$. Associative follows readily.

Security. We check security properties as follows:

(orthogonality.) For $g_1^{\mathbf{D}}$ and $g_2^{\mathbf{B}^* \mathbf{e}_{2d}}$, we have

$$\mu(g_2^{\mathbf{B}^* \mathbf{e}_{2d}}) = e(g_1^{\mathbf{D}}, g_2^{\mathbf{B}^* \mathbf{e}_{2d}}) = (1, \ldots, 1)^\top,$$

where in the equality, we use the fact that

$$\mathbf{D}^\top (\mathbf{B}^* \mathbf{e}_{2d}) = \pi_L(\mathbf{B})^\top (\mathbf{B}^* \mathbf{e}_{2d}) = (0, \ldots, 0)^\top.$$

(non-degeneracy.) For all $g_1^{\mathbf{F}\hat{\mathbf{s}}} \leftarrow \widehat{\mathsf{SampG}}_0(\mathrm{PP}, \mathrm{SP}; \hat{\mathbf{s}})$, we have

$$e(g_1^{\mathbf{F}\hat{\mathbf{s}}}, g_2^{\mathbf{B}^*\mathbf{e}_{2d}}) = e(g_1, g_2)^{\mathbf{e}_d^\top \hat{\mathbf{s}}} \neq 1,$$

whenever $\mathbf{e}_d^\top \hat{\mathbf{s}} \neq 0$, which occurs with probability $1 - 1/p$ over $\hat{\mathbf{s}}$; thus, $e(g_1^{\mathbf{F}\hat{\mathbf{s}}}, g_2^{\mathbf{B}^*\mathbf{e}_{2d}})^\alpha$ is identically distributed to the uniform distribution over G_T with probability $1 - 2^{-\Omega(\lambda)}$, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$.

($\mathbb{H}$-subgroup.) This follows readily from the fact that $\mathbb{Z}_p^{2d}$ is an additive group.

We establish left subgroup and nested-hiding indistinguishability in next three subsections, under the d-LIN assumption in prime-order groups.

6.3 Left Subgroup Indistinguishability

We may rewrite the corresponding advantage function as:

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

where

$$\begin{aligned}
&(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\
&\mathbf{g} := (g_1^{\mathbf{D}\mathbf{s}}, g_1^{\mathbf{D}_1\mathbf{s}}, \ldots, g_1^{\mathbf{D}_n\mathbf{s}}),\ \mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d;\\
&\hat{\mathbf{g}} := (g_1^{\mathbf{F}\hat{\mathbf{s}}}, g_1^{\mathbf{F}_1\hat{\mathbf{s}}}, \ldots, g_1^{\mathbf{F}_n\hat{\mathbf{s}}}),\ \hat{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d.
\end{aligned}$$

Lemma 8 (d-LIN to LS). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{LS}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{d\text{-LIN}}_{\mathcal{B}}(\lambda).$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + d^2 \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We may write $(\mathrm{PP}, \mathbf{g}, \mathbf{g} \cdot \hat{\mathbf{g}})$ in term of $\mathbf{B}, \mathbf{B}^*, \mathbf{A}_1, \ldots, \mathbf{A}_n, \mathbf{R}$ as follows:

$$\mathrm{PP} := \left((p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); \begin{matrix} g_1^{\pi_L(\mathbf{B})}, g_1^{\pi_L(\mathbf{B}\mathbf{A}_1)}, \ldots, g_1^{\pi_L(\mathbf{B}\mathbf{A}_n)} \\ g_2^{\mathbf{B}^*\mathbf{R}},\ g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}},\ \ldots, g_2^{\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R}} \end{matrix} \right),$$

and

$$\begin{aligned}
\mathbf{g} &:= (g_1^{\mathbf{B}\binom{\mathbf{s}}{\mathbf{0}}}, g_1^{\mathbf{B}\mathbf{A}_1\binom{\mathbf{s}}{\mathbf{0}}}, \ldots, g_1^{\mathbf{B}\mathbf{A}_n\binom{\mathbf{s}}{\mathbf{0}}}),\\
\mathbf{g} \cdot \hat{\mathbf{g}} &:= (g_1^{\mathbf{B}\binom{\mathbf{s}}{\hat{\mathbf{s}}}}, g_1^{\mathbf{B}\mathbf{A}_1\binom{\mathbf{s}}{\hat{\mathbf{s}}}}, \ldots, g_1^{\mathbf{B}\mathbf{A}_n\binom{\mathbf{s}}{\hat{\mathbf{s}}}}),
\end{aligned}$$

where $\mathbf{s}, \hat{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ (and thus $\binom{\mathbf{s}}{\mathbf{0}}, \binom{\mathbf{s}}{\hat{\mathbf{s}}} \in \mathbb{Z}_p^{2d}$).

The adversary $\mathcal{B}$ gets as input

$$\left((p, G_1, G_2, G_T, e); g_1, g_2, g_1^{a_1}, \ldots, g_1^{a_d}, g_1^{a_{d+1}}, g_1^{a_1 s_1}, \ldots, g_1^{a_d s_d}, g_1^{a_{d+1}(s_1 + \ldots + s_d) + s_{d+1}} \right),$$

where either $s_{d+1} = 0$ or $s_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, and proceeds as follows:

Programming $\mathbf{s}, \hat{\mathbf{s}}$. $\mathcal{B}$ picks $\gamma_1, \ldots, \gamma_d \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and implicitly sets

$$\mathbf{s} := (s_1, \ldots, s_d)^\top \quad \text{and} \quad \hat{\mathbf{s}} := (s_{d+1}\gamma_1, \ldots, s_{d+1}\gamma_d)^\top$$

where $(s_1, \ldots, s_d, s_{d+1})$ are as defined in its input. Later on, $\mathcal{B}$ will output $\mathbf{g}$ if $s_{d+1} = 0$ and $\hat{\mathbf{s}} = \mathbf{0}$ and $\mathbf{g} \cdot \hat{\mathbf{g}}$ if $s_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and $\hat{\mathbf{s}} = (s_{d+1}\gamma_1, \ldots, s_{d+1}\gamma_d)^\top$.

Programming $\mathbf{B}, \mathbf{B}^*, \mathbf{A}_1, \ldots, \mathbf{A}_n, \mathbf{R}$. We define

$$\mathbf{W} := \begin{pmatrix} a_1 & & & & & & \\ & a_2 & & & & & \\ & & \ddots & & & & \\ & & & a_d & & & \\ \gamma_1 a_{d+1} & \gamma_1 a_{d+1} & \cdots & \gamma_1 a_{d+1} & 1 & & \\ \gamma_2 a_{d+1} & \gamma_2 a_{d+1} & \cdots & \gamma_2 a_{d+1} & & 1 & \\ \vdots & \vdots & & \vdots & & & \ddots \\ \gamma_d a_{d+1} & \gamma_d a_{d+1} & \cdots & \gamma_d a_{d+1} & & & 1 \end{pmatrix},$$

and set $\mathbf{W}^* = (\mathbf{W}^{-1})^\top$. It follows immediately that

$$g_1^{\mathbf{W}\binom{\mathbf{s}}{\hat{\mathbf{s}}}} = \begin{pmatrix} g_1^{a_1 s_1} \\ \vdots \\ g_1^{a_d s_d} \\ g_1^{\gamma_1(a_{d+1}(s_1+\cdots+s_d)+s_{d+1})} \\ \vdots \\ g_1^{\gamma_d(a_{d+1}(s_1+\cdots+s_d)+s_{d+1})} \end{pmatrix}.$$

Next, $\mathcal{B}$ samples $\tilde{\mathbf{B}}, \tilde{\mathbf{R}} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p)$, along with $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(2d)\times(2d)}$, and implicitly sets:

$$\begin{aligned} \tilde{\mathbf{B}}^* &:= (\tilde{\mathbf{B}}^{-1})^\top; \\ \mathbf{R} &:= \mathbf{W}^\top \tilde{\mathbf{R}}; \\ (\mathbf{B}, \mathbf{B}^*) &:= (\tilde{\mathbf{B}}\mathbf{W}, \tilde{\mathbf{B}}^*\mathbf{W}^*); \\ \mathbf{A}_i &:= \mathbf{W}^{-1}\tilde{\mathbf{A}}_i\mathbf{W}. \end{aligned}$$

It is clear that $(\mathbf{B}, \mathbf{B}^*)$, $\mathbf{A}_1, \ldots, \mathbf{A}_n$, and $\mathbf{R}$ are properly distributed and completely hide $(\gamma_1, \ldots, \gamma_d)$ (so that $\hat{\mathbf{s}}$ is uniformly distributed over $\mathbb{Z}_p^d$ whenever $s_{d+1} \neq 0$). Moreover, we have:

$$\begin{aligned} \mathbf{B}\mathbf{A}_i = (\tilde{\mathbf{B}}\mathbf{W})(\mathbf{W}^{-1}\tilde{\mathbf{A}}_i\mathbf{W}) = \tilde{\mathbf{B}}\tilde{\mathbf{A}}_i\mathbf{W} \quad \text{and} \quad \mathbf{B}^*\mathbf{R} = (\tilde{\mathbf{B}}^*\mathbf{W}^*)(\mathbf{W}^\top\tilde{\mathbf{R}}) = \tilde{\mathbf{B}}^*\tilde{\mathbf{R}} \\ \text{and} \quad \mathbf{B}^*\mathbf{A}_i^\top\mathbf{R} = (\tilde{\mathbf{B}}^*\mathbf{W}^*)(\mathbf{W}^\top\tilde{\mathbf{A}}_i^\top(\mathbf{W}^{-1})^\top)(\mathbf{W}^\top\tilde{\mathbf{R}}) = \tilde{\mathbf{B}}^*\tilde{\mathbf{A}}_i^\top\tilde{\mathbf{R}}. \end{aligned}$$

Simulating PP. Observe that for all $i = 1, \ldots, n$, $\mathcal{B}$ can compute

$$\begin{aligned} g_1^{\pi_L(\mathbf{B})} &= g_1^{\pi_L(\tilde{\mathbf{B}}\mathbf{W})} \quad \text{and} \quad g_1^{\pi_L(\mathbf{B}\mathbf{A}_i)} = g_1^{\pi_L(\tilde{\mathbf{B}}\tilde{\mathbf{A}}_i\mathbf{W})} \\ g_2^{\mathbf{B}^*\mathbf{R}} &= g_2^{\tilde{\mathbf{B}}^*\tilde{\mathbf{R}}} \quad \text{and} \quad g_2^{\mathbf{B}^*\mathbf{A}_i^\top\mathbf{R}} = g_2^{\tilde{\mathbf{B}}^*\tilde{\mathbf{A}}_i^\top\tilde{\mathbf{R}}} \end{aligned}$$

since it knows $(\tilde{\mathbf{B}}, \tilde{\mathbf{B}}^*, \tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_n), g_1^{\mathbf{W}}$ as well as $\tilde{\mathbf{R}}$.

Simulating the challenge. $\mathcal{B}$ outputs the challenge as

$$g_1^{\mathbf{B}\binom{\mathbf{s}}{\hat{\mathbf{s}}}} = g_1^{\tilde{\mathbf{B}}\mathbf{W}\binom{\mathbf{s}}{\hat{\mathbf{s}}}} = g_1^{\tilde{\mathbf{B}}\begin{pmatrix} a_1 s_1 \\ \vdots \\ a_d s_d \\ \gamma_1(a_{d+1}(s_1+\cdots+s_d)+s_{d+1}) \\ \vdots \\ \gamma_d(a_{d+1}(s_1+\cdots+s_d)+s_{d+1}) \end{pmatrix}}$$

along with

$$g_1^{\mathbf{B}\mathbf{A}_i\binom{\mathbf{s}}{\hat{\mathbf{s}}}} = g_1^{\tilde{\mathbf{B}}\tilde{\mathbf{A}}_i\mathbf{W}\binom{\mathbf{s}}{\hat{\mathbf{s}}}} = g_1^{\tilde{\mathbf{B}}\tilde{\mathbf{A}}_i\begin{pmatrix} a_1 s_1 \\ \vdots \\ a_d s_d \\ \gamma_1(a_{d+1}(s_1+\cdots+s_d)+s_{d+1}) \\ \vdots \\ \gamma_d(a_{d+1}(s_1+\cdots+s_d)+s_{d+1}) \end{pmatrix}} \qquad i = 1, \ldots, n$$

Observe that if $s_{d+1} = 0$, then $\hat{\mathbf{s}} = \mathbf{0}$ and the output challenge equals $\mathbf{g}$. On the other hand, if $s_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, then $\hat{\mathbf{s}} = (s_{d+1}\gamma_1, \ldots, s_{d+1}\gamma_d)^\top$ is uniformly distributed over $\mathbb{Z}_p^d$ and the output challenge equals $\mathbf{g} \cdot \hat{\mathbf{g}}$.

The lemma then follows readily. □

6.4 Many-Tuple Lemma

We want to prove a many-tuple lemma which will be used in the proofs. This lemma is implicit in [25, 22] (see [22, Lemma 2]):

Lemma 9. *There exists an efficient algorithm that on input 1^q, a group G and*

$$(\ g, g^{a_1}, \ldots, g^{a_d}, g^{a_{d+1}}, g^{a_1 r_1}, \ldots, g^{a_d r_d}, g^{a_{d+1}(r_1+\ldots+r_d)+r_{d+1}}\ ) \in G^{2d+3},$$

where either $r_{d+1} = 0$ or $r_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, outputs $(g^{\mathbf{VZ}}, g^{\mathbf{Z}})$ for some matrix $\mathbf{V} \in \mathbb{Z}_p^{d\times d}$, along with q tuples

$$(g^{\hat{\mathbf{r}}_j}, g^{\mathbf{t}_j}),\ j = 1, \ldots, q,$$

where

$$\mathbf{t}_j = \begin{cases} \mathbf{V}\hat{\mathbf{r}}_j & \text{if } r_{d+1} = 0 \\ \mathbf{V}\hat{\mathbf{r}}_j + \gamma_j \mathbf{e}_d & \text{if } r_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^* \end{cases}$$

and $\hat{\mathbf{r}}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d, \gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and $\mathbf{Z}$ is an invertible diagonal matrix.

Proof. The algorithm proceeds as follows:

Defining $\mathbf{V}, \mathbf{Z}$. We implicitly define

$$\mathbf{V} := \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ r_1 & \cdots & r_{d-1} & r_d \end{pmatrix} \in \mathbb{Z}_p^{d\times d},\ \mathbf{Z} := \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_d \end{pmatrix} \in \mathbb{Z}_p^{d\times d},\ \mathbf{P} := \begin{pmatrix} a_1 & & a_{d+1} \\ & \ddots & \vdots \\ & a_d & a_{d+1} \end{pmatrix} \in \mathbb{Z}_p^{d\times(d+1)}.$$

Clearly, we can compute $g^{\mathbf{VZ}}$, $g^{\mathbf{Z}}$, and $g^{\mathbf{P}}$. In addition, we can compute

$$g^{\mathbf{C}} := \begin{pmatrix} g^{a_1} & & & g^{a_{d+1}} \\ & \ddots & & \vdots \\ & & g^{a_{d-1}} & g^{a_{d+1}} \\ g^{a_1 r_1} & \cdots & g^{a_{d-1}r_{d-1}} & g^{a_d r_d}\ g^{a_{d+1}(r_1+\cdots+r_d)+r_{d+1}} \end{pmatrix} \in G^{d\times(d+1)}.$$

Observe that $g^{\mathbf{C}} = g^{\mathbf{VP}}$ if $r_{d+1} = 0$.

Generating q tuples. For $j = 1, \ldots, q$, we pick $\tilde{\mathbf{r}}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{d+1}$ and output

$$(g^{\hat{\mathbf{r}}_j}, g^{\mathbf{t}_j}) := \left(g^{\mathbf{P}\tilde{\mathbf{r}}_j},\ g^{\mathbf{C}\tilde{\mathbf{r}}_j} \right),$$

where we have implicitly set $\hat{\mathbf{r}}_j := \mathbf{P}\tilde{\mathbf{r}}_j$ and $\mathbf{t}_j := \mathbf{C}\tilde{\mathbf{r}}_j$. It is easy to see that $\hat{\mathbf{r}}_j$ is uniformly distributed over $\mathbb{Z}_p^d$.

It remains to analyze the distribution of the q tuples:

- If $r_{d+1} = 0$, we have $\mathbf{C} = \mathbf{VP}$ and thus $\mathbf{t}_j = \mathbf{V}\hat{\mathbf{r}}_j$. (Alternatively, this may be viewed a special case corresponding to $r_{d+1} = 0$ and thus $\gamma_j = 0$.)

- If $r_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, we have implicitly set $\gamma_j := r_{d+1}\mathbf{e}_{d+1}^\top \tilde{\mathbf{r}}_j$, and thus $\mathbf{t}_j = \mathbf{V}\hat{\mathbf{r}}_j + \gamma_j \mathbf{e}_{d+1}$. A simple calculation shows that

$$\begin{pmatrix} \hat{\mathbf{r}}_j \\ \gamma_j \end{pmatrix} = \begin{pmatrix} a_1 & & & a_{d+1} \\ & \ddots & & \vdots \\ & & a_d & a_{d+1} \\ 0 & \ldots & 0 & r_{d+1} \end{pmatrix} \tilde{\mathbf{r}}_j$$

Moreover, whenever $a_1, \ldots, a_d, r_{d+1} \neq 0$, the matrix on the right has full rank, and thus $(\tilde{\mathbf{r}}_j, \gamma_j)$ is properly distributed.

The lemma then follows readily. □

6.5 Nested-Hiding Indistinguishability

We may rewrite the corresponding advantage function as:

$$\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda, q) := \max_{i\in[n]} \left|\Pr[\ \mathcal{A}(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-i}, \boxed{\mathbf{h}^1, \ldots, \mathbf{h}^q}) = 1\ ] - \Pr[\ \mathcal{A}(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-i}, \boxed{\mathbf{h}'^1, \ldots, \mathbf{h}'^q}) = 1\ ]\right|$$

where

$$\begin{aligned}
&(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\
&\hat{\mathbf{g}} := (g_1^{\mathbf{F}\hat{\mathbf{s}}},\ g_1^{\mathbf{F}_1\hat{\mathbf{s}}},\ \ldots,\ g_1^{\mathbf{F}_n\hat{\mathbf{s}}});\ \hat{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d;\\
&\mathbf{h}^j := (g_2^{\mathbf{D}^*\mathbf{r}_j}, g_2^{\mathbf{D}_1^*\mathbf{r}_j}, \ldots, \boxed{g_2^{\mathbf{D}_i^*\mathbf{r}_j}}, \ldots, g_2^{\mathbf{D}_n^*\mathbf{r}_j});\ \mathbf{r}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d};\ j = 1, \ldots, q;\\
&\mathbf{h}'^j := (g_2^{\mathbf{D}^*\mathbf{r}_j}, g_2^{\mathbf{D}_1^*\mathbf{r}_j}, \ldots, \boxed{g_2^{\mathbf{D}_i^*\mathbf{r}_j} \cdot (h^*)^{\gamma_j}}, \ldots, g_2^{\mathbf{D}_n^*\mathbf{r}_j});\ \gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p;\ j = 1, \ldots, q.
\end{aligned}$$

Lemma 10 (d-LIN to NS). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda, q) \le \mathsf{Adv}^{d\text{-LIN}}_{\mathcal{B}}(\lambda).$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + d^2 q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We note that it is sufficient to bound the following advantage function for any adversary $\mathcal{A}$:

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{NS}'}_{\mathcal{A}}(\lambda) := \Big|&\Pr[\ \mathcal{A}((p, G_1, G_2, e); g_1, g_2, g_1^{\pi_L(\mathbf{A})}, g_2^{\mathbf{R}}, g_2^{\mathbf{A}^\top\mathbf{R}}, g_2^{\mathbf{R}\mathbf{r}_1}, \ldots, g_2^{\mathbf{R}\mathbf{r}_q}, \boxed{g_2^{\mathbf{u}_1}, \ldots, g_2^{\mathbf{u}_q}}) = 1\ ] -\\
&\Pr[\ \mathcal{A}((p, G_1, G_2, e); g_1, g_2, g_1^{\pi_L(\mathbf{A})}, g_2^{\mathbf{R}}, g_2^{\mathbf{A}^\top\mathbf{R}}, g_2^{\mathbf{R}\mathbf{r}_1}, \ldots, g_2^{\mathbf{R}\mathbf{r}_q}, \boxed{g_2^{\mathbf{u}_1+\gamma_1\mathbf{e}_{2d}}, \ldots, g_2^{\mathbf{u}_q+\gamma_q\mathbf{e}_{2d}}}) = 1\ ]\Big|
\end{aligned}$$

where

$$\begin{aligned}
&(p, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda);\\
&\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(2d)\times(2d)},\ \mathbf{R} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p);\\
&\gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p,\ \mathbf{r}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d},\ \mathbf{u}_j := \mathbf{A}^\top\mathbf{R}\mathbf{r}_j, \quad j = 1, \ldots, q.
\end{aligned}$$

We defer the proof of the claim for now, and first explain how the lemma follows from the claim. We sample $\mathbf{B} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p)$, $\mathbf{A}_1, \ldots, \mathbf{A}_{i-1}, \mathbf{A}_{i+1}, \ldots, \mathbf{A}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(2d)\times(2d)}$, $\hat{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and implicitly set $\mathbf{A}_i := \mathbf{A}$. Observe that:

$$\mathrm{PP} := \left((p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e);\ \begin{matrix} g_1^{\pi_L(\mathbf{B})},\ g_1^{\pi_L(\mathbf{B}\mathbf{A}_1)},\ \ldots,\ g_1^{\pi_L(\mathbf{B}\mathbf{A}_n)} \\ g_2^{\mathbf{B}^*\mathbf{R}},\ g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}},\ \ldots,\ g_2^{\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R}} \end{matrix} \right), \quad h^* := g_2^{\mathbf{B}^*\mathbf{e}_{2d}},$$

and

$$\begin{aligned}
\hat{\mathbf{g}}_{-i} &:= (g_1^{\pi_R(\mathbf{B})\hat{\mathbf{s}}}, g_1^{\pi_R(\mathbf{B}\mathbf{A}_1)\hat{\mathbf{s}}}, \ldots, g_1^{\pi_R(\mathbf{B}\mathbf{A}_{i-1})\hat{\mathbf{s}}}, g_1^{\pi_R(\mathbf{B}\mathbf{A}_{i+1})\hat{\mathbf{s}}}, \ldots, g_1^{\pi_R(\mathbf{B}\mathbf{A}_n)\hat{\mathbf{s}}})\\
\mathbf{h}^j &:= (g_2^{\mathbf{B}^*\mathbf{R}\mathbf{r}_j}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}\mathbf{r}_j}, \ldots, \boxed{g_2^{\mathbf{B}^*\mathbf{A}_i^\top\mathbf{R}\mathbf{r}_j}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R}\mathbf{r}_j})\\
\mathbf{h}'^j &:= (g_2^{\mathbf{B}^*\mathbf{R}\mathbf{r}_j}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}\mathbf{r}_j}, \ldots, \boxed{g_2^{\mathbf{B}^*\mathbf{A}_i^\top\mathbf{R}\mathbf{r}_j} \cdot g_2^{\gamma_j\mathbf{B}^*\mathbf{e}_{2d}}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R}\mathbf{r}_j}),
\end{aligned}$$

– we can simulate PP and h^* since we know $\mathbf{B}, \mathbf{A}_1, \ldots, \mathbf{A}_{i-1}, \mathbf{A}_{i+1}, \ldots, \mathbf{A}_n$ and $g_1^{\pi_L(\mathbf{A})}, g_2^{\mathbf{R}}, g_2^{\mathbf{A}^\top\mathbf{R}}$;

– we can simulate $\hat{\mathbf{g}}_{-i}$ since we know $\mathbf{B}, \mathbf{A}_1, \ldots, \mathbf{A}_{i-1}, \mathbf{A}_{i+1}, \ldots, \mathbf{A}_n$ and $\hat{\mathbf{s}}$;
– given either $g_2^{\mathbf{u}_1}, \ldots, g_2^{\mathbf{u}_q}$ or $g_2^{\mathbf{u}_1+\gamma_1\mathbf{e}_{2d}}, \ldots, g_2^{\mathbf{u}_q+\gamma_q\mathbf{e}_{2d}}$, we can simulate either $\mathbf{h}_1, \ldots, \mathbf{h}_q$ or $\mathbf{h}'_1, \ldots, \mathbf{h}'_q$ respectively since we know $\mathbf{B}, \mathbf{A}_1, \ldots, \mathbf{A}_{i-1}, \mathbf{A}_{i+1}, \ldots, \mathbf{A}_n, g_2^{\mathbf{R}\mathbf{r}_1}, \ldots, g_2^{\mathbf{R}\mathbf{r}_q}$.

The lemma then follows readily. □

Proof (of claim). We use $\mathbf{0}$ to denote the all zeroes vector in $\mathbb{Z}_p^d$, and $\mathbf{0}_d$ to denote the all zeroes matrix in $\mathbb{Z}_p^{d\times d}$. The adversary $\mathcal{B}$ gets as input

$$\left((p, G_1, G_2, G_T, e); g_1, g_2, g_2^{a_1}, \ldots, g_2^{a_d}, g_2^{a_{d+1}}, g_2^{a_1 r_1}, \ldots, g_2^{a_d r_d}, g_2^{a_{d+1}(r_1+\ldots+r_d)+r_{d+1}} \right),$$

where either $r_{d+1} = 0$ or $r_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, and proceeds as follows:

Generating q tuples. Run the algorithm in Lemma 9 on input 1^q, the group G_2 and the DLIN-tuple to obtain

$$(g_2^{\mathbf{V}\mathbf{Z}}, g^{\mathbf{Z}}) \quad \text{and} \quad (g_2^{\hat{\mathbf{r}}_j}, g_2^{\mathbf{t}_j}),\ j = 1, \ldots, q.$$

Programming A. Sample $\tilde{\mathbf{A}} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{2d\times 2d}$ and implicitly set:

$$\mathbf{A} := \tilde{\mathbf{A}} + \begin{pmatrix} \mathbf{0}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{V}^\top \end{pmatrix}$$

Note that $g_1^{\pi_L(\mathbf{A})} = g_1^{\pi_L(\tilde{\mathbf{A}})}$.

Programming R. We pick $\tilde{\mathbf{R}} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p)$ and implicitly set

$$\mathbf{R} := \begin{pmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{Z} \end{pmatrix} \tilde{\mathbf{R}}.$$

Observe that $\mathbf{R}$ is properly distributed as long as $a_1, \ldots, a_d \neq 0$. Moreover, we can compute $g_2^{\mathbf{R}}$ since $\tilde{\mathbf{R}}$ and $g_2^{\mathbf{Z}}$ are known.

Simulating $g_2^{\mathbf{A}^\top\mathbf{R}}$. We can write $\mathbf{A}^\top\mathbf{R}$ as

$$\mathbf{A}^\top\mathbf{R} = \tilde{\mathbf{A}}^\top\mathbf{R} + \begin{pmatrix} \mathbf{0}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{V} \end{pmatrix} \begin{pmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{Z} \end{pmatrix} \tilde{\mathbf{R}} = \tilde{\mathbf{A}}^\top\mathbf{R} + \begin{pmatrix} \mathbf{0}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{V}\mathbf{Z} \end{pmatrix} \tilde{\mathbf{R}}.$$

Observe that we can compute $g_2^{\mathbf{A}^\top\mathbf{R}}$ since we know $(\tilde{\mathbf{A}}, g_2^{\mathbf{R}})$ and $(g_2^{\mathbf{V}\mathbf{Z}}, \tilde{\mathbf{R}})$.

Simulating $g_2^{\mathbf{R}\mathbf{r}_1}, \ldots, g_2^{\mathbf{R}\mathbf{r}_q}$. For $j = 1, \ldots, q$, we pick $\tilde{\mathbf{r}}_j \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and implicitly set

$$\mathbf{r}_j := \mathbf{R}^{-1} \begin{pmatrix} \tilde{\mathbf{r}}_j \\ \hat{\mathbf{r}}_j \end{pmatrix}.$$

Observe that $\mathbf{r}_j$ is properly distributed as long as $\mathbf{R} \in \mathsf{GL}_{2d}(\mathbb{Z}_p)$. In addition, we can compute $g_2^{\mathbf{R}\mathbf{r}_j}$ since we know $\tilde{\mathbf{r}}_j$ and $g_2^{\hat{\mathbf{r}}_j}$.

Simulating the challenge. For $j = 1, \ldots, q$, observe that

$$\mathbf{A}^\top \mathbf{R}\mathbf{r}_j = \tilde{\mathbf{A}}^\top(\mathbf{R}\mathbf{r}_j) + \begin{pmatrix} \mathbf{0}_d \, \mathbf{0}_d \\ \mathbf{0}_d \, \mathbf{V} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{r}}_j \\ \hat{\mathbf{r}}_j \end{pmatrix} = \tilde{\mathbf{A}}^\top(\mathbf{R}\mathbf{r}_j) + \begin{pmatrix} \mathbf{0} \\ \mathbf{V}\hat{\mathbf{r}}_j \end{pmatrix}.$$

Note that we can compute $g_2^{\tilde{\mathbf{A}}^\top(\mathbf{R}\mathbf{r}_j)}$ since we know $(\tilde{\mathbf{A}}, g_2^{\mathbf{R}\mathbf{r}_j})$. Now we replace $g_2^{\mathbf{V}\hat{\mathbf{r}}_j}$ with $g_2^{\mathbf{t}_j}$, and output the j'th challenge as

$$g_2^{\tilde{\mathbf{A}}^\top(\mathbf{R}\mathbf{r}_j)} \cdot \begin{pmatrix} g_2^{\mathbf{0}} \\ g_2^{\mathbf{t}_j} \end{pmatrix}.$$

Observe that:

- if $r_{d+1} = 0$, we have $\mathbf{t}_j = \mathbf{V}\hat{\mathbf{r}}_j$ and therefore the j'th output challenge equals $g_2^{\mathbf{A}^\top \mathbf{R}\mathbf{r}_j}$.
- if $r_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, we have $\mathbf{t}_j = \mathbf{V}\hat{\mathbf{r}}_j + \gamma_j \mathbf{e}_d$ and therefore the j'th output challenge equals $g_2^{\mathbf{A}^\top \mathbf{R}\mathbf{r}_j + \gamma_j \mathbf{e}_{2d}}$.

The claim then follows readily. □

7 Concrete IBE Scheme from d-LIN in Prime-Order Groups

In this section, we provide a self-contained description of the IBE scheme under the d-LIN assumption 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

$$\mathrm{MPK} := \left(g_1^{\pi_L(\mathbf{B})}, g_1^{\pi_L(\mathbf{B}\mathbf{A}_1)}, \ldots, g_1^{\pi_L(\mathbf{B}\mathbf{A}_{2n})}; e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})} \right) \in (G_1^{2d\times d})^{2n+1} \times G_T^d$$

$$\mathrm{MSK} := \left(g_2^{\mathbf{k}}, g_2^{\mathbf{B}^*\mathbf{R}}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top \mathbf{R}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_{2n}^\top \mathbf{R}} \right) \in G_2^{2d} \times (G_2^{2d\times 2d})^{2n+1}$$

Note that g_2-components in PP are not required for encryption, we moved them into MSK.

$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m)$: 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

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

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$: 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

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

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

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

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] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC*, pages 325–341, 2005.

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

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

[11] J. Chen and H. Wee. Fully, (almost) tightly secure ibe and dual system groups. In *CRYPTO (2)*, pages 435–460, 2013.

[12] J. Chen and H. Wee. Dual system groups and its applications — compact HIBE and more. Full version in preparation, 2013.

[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] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

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

[22] 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.

[23] 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.

[24] 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.

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

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

[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 system encryption via predicate encodings. In *TCC*, 2014. To appear.