Dual System Groups and its Applications — Compact HIBE and More*

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

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

Abstract. We introduce the notion of *dual system groups*.

- We show how to derive compact HIBE by instantiating the dual system framework in Waters (Crypto '09) and Lewko and Waters (TCC '10) with dual system groups. Our construction provides a unified treatment of the prior compact HIBE schemes from static assumptions.
- We show how to instantiate dual system groups under the decisional subgroup assumption in composite-order groups and the decisional linear assumption (d-LIN) in prime-order groups. Along the way, we provide new tools for simulating properties of composite-order bilinear groups in prime-order groups. In particular, we present new randomization and parameter-hiding techniques in prime-order groups.

Combining the two, we obtain a number of new encryption schemes, notably

- a new construction of IBE in prime-order groups with shorter parameters;
- a new construction of compact HIBE in prime-order groups whose structure closely mirrors the selectively secure HIBE scheme of Boneh, Boyen and Goh (Eurocrypt '05);
- a new construction of compact spatial encryption in prime-order groups.

Table of Contents

1 Introduction 1
2 Preliminaries 5
3 Dual System Groups 6
4 Compact HIBE from Dual System Groups 10
5 Instantiations in composite-order groups 18
6 Instantiations from d-LIN in prime-order groups 22
7 Concrete HIBE scheme from d-LIN in prime-order groups 30
8 Spatial Encryption from Dual System Groups 31

* A preliminary version of this work appeared as a merge with [5] at CRYPTO 2013 [4].

** 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: wee@di.ens.fr. Supported in part by NSF Awards CNS-1237429 and CNS-1319021. Work done while at George Washington University and while visiting NTU.

1 Introduction

The current assumptions used for pairings-based functional encryption may be broadly classified into two categories: those pertaining to prime-order groups and those pertaining to composite-order groups. From a theoretical stand-point, it is often easier to design schemes in composite-order groups. However, from the practical stand-point, prime-order groups are preferable as they admit more efficient and compact instantiations. Since the elliptic curve group order must be infeasible to factor, it must be at least (say) 1024 bits. On the other hand, a 160-bit prime-order elliptic curve group provides an equivalent level of security (per NIST recommendations in NIST SP 800-57). More generally, group operations and especially pairing computations are substantially slower on composite-order curves for the same level of security. To mitigate the gap between theoretical design and practical efficiency, a series of works demonstrated general techniques for converting cryptosystems relying on composite-order groups to cryptosystems based on prime-order groups [13, 14, 7, 10, 6].

However, there are still cases where these transformations do not cover, notably, the composite-order HIBE with constant-size ciphertext in [11]. Naively applying previous transformations yield a prime-order scheme with linear-size ciphertext, that is, they do not preserve the parameters (e.g. ciphertext size) of the original composite-order scheme. We note that there is a more ad-hoc transformation given in [16] which yields HIBE with constant-size ciphertext in prime-order groups. Our goal is to find more general "parameter-preserving" tools for simulating composite-order bilinear groups in the prime order setting; such tools would shed new theoretical insight into the design of efficient functional encryption schemes.

1.1 Our contributions

We introduce a novel notion of dual system groups. Our main results are as follows:

- a generic construction of compact HIBE from dual system groups similar to the Lewko-Waters scheme over composite-order groups [11]; and
- instantiations of dual system groups under the d-LIN assumption in prime-order bilinear groups and the subgroup decisional assumption in composite-order bilinear groups respectively.

Along the way, we provide new tools for simulating properties of composite-order bilinear groups in prime-order groups. In particular, we present new randomization and parameter-hiding techniques in prime-order groups.

Putting the two together, we obtain a new construction of compact HIBE in prime-order groups, as well as new insights into the structural properties needed for Waters' dual system encryption methodology [19]. We compare our schemes with prior constructions in Tables 1 and 2. In particular, our compact prime-order HIBE improves upon the efficiency of the prior scheme in [16] by a factor of two in nearly all parameters, and has an arguably much simpler description. We note that we even improve upon prior constructions of IBE in prime-order groups.[1]

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

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

[1] A subsequent work [9] achieves incomparable efficiency guarantees.

Reference	$\lvert\mathsf{MPK}\rvert$	$\lvert\mathsf{SK}\rvert$	$\lvert\mathsf{CT}\rvert$	T_{KeyGen}	T_{Enc}	T_{Dec}	assumption
Wat05 [18]	$(\lambda+4)\lvert G_1\rvert$	$2\lvert G_2\rvert$	$2\lvert G_1\rvert+\lvert G_T\rvert$	$2E_2$	$2E_1+E_T$	$2P$	DBDH
Wat09 [19]	$13\lvert G_1\rvert+\lvert G_T\rvert$	$8\lvert G_2\rvert+\lvert\mathbb{Z}_p\rvert$	$9\lvert G_1\rvert+\lvert G_T\rvert+\lvert\mathbb{Z}_p\rvert$	$8E_2$	$14E_1+E_T$	$9P+E_T$	DLIN
Lew12 [10]	$24\lvert G_1\rvert+\lvert G_T\rvert$	$6\lvert G_2\rvert$	$6\lvert G_1\rvert+\lvert G_T\rvert$	$6E_2$	$24E_1+E_T$	$6P$	DLIN
RCS12 [17]	$9\lvert G_1\rvert+\lvert G_T\rvert$	$6\lvert G_2\rvert+\lvert\mathbb{Z}_p\rvert$	$7\lvert G_1\rvert+\lvert G_T\rvert+\lvert\mathbb{Z}_p\rvert$	$6E_2$	$10E_1+E_T$	$7P+E_T$	XDH + DLIN
CLL+12 [6]	$8\lvert G_1\rvert+\lvert G_T\rvert$	$4\lvert G_2\rvert$	$4\lvert G_1\rvert+\lvert G_T\rvert$	$4E_2$	$8E_1+E_T$	$4P$	SXDH
JR13 [9]	$5\lvert G_1\rvert+\lvert G_T\rvert$	$5\lvert G_2\rvert$	$3\lvert G_1\rvert+\lvert G_T\rvert+\lvert\mathbb{Z}_p\rvert$	$5E_2$	$5E_1+E_T$	$3P+2E_2$	SXDH
Ours	$6\lvert G_1\rvert+\lvert G_T\rvert$	$4\lvert G_2\rvert$	$4\lvert G_1\rvert+\lvert G_T\rvert$	$4E_2$	$6E_1+E_T$	$4P$	SXDH
(Sec 7)	$18\lvert G_1\rvert+2\lvert G_T\rvert$	$6\lvert G_2\rvert$	$6\lvert G_1\rvert+\lvert G_T\rvert$	$6E_2$	$18E_1+2E_T$	$6P$	DLIN

Fig. 1. Comparison amongst IBE schemes based on asymmetric bilinear groups of prime order p with pairing $e : G_1 \times G_2 \to G_T$ and security parameter λ, where (E_1, E_2, E_T, P) denote G_1-exponentiation, G_2-exponentiation, G_T-exponentiation and a pairing respectively. For KeyGen, we assume that we store exponents instead of group elements in MSK. Here, we omitted the G_2 terms in MPK in our scheme, which are not needed for the correctness of the scheme.

Reference	$\lvert\mathsf{CT}\rvert$	T_{KeyGen}	T_{Enc}	T_{Dec}	assumption
BBG05 [2]	$2\lvert G_1\rvert+\lvert G_T\rvert$	$(n+1)E_2$	$(n+2)E_1+E_T$	$2P$	n-DBDHE
Wat09 [19]	$(n+8)\lvert G_1\rvert+\lvert G_T\rvert$	$(2n+7)E_2$	$(3n+11)E_1+E_T$	$(2n+7)P+nE_T$	DLIN
LW10 [11]	$2\lvert G_N\rvert+\lvert G_T\rvert$	$(n+1)E_N$	$(n+2)E_N+E_T$	$2P$	composite
OT10 [15]	$(7n+5)\lvert G_1\rvert+\lvert G_T\rvert$	$(7n+5)E_2$	$(21n+15)E_1+E_T$	$(7n+5)P$	DLIN
OT11 [16]	$13\lvert G_1\rvert+\lvert G_T\rvert$	$(16n-3)E_2$	$(8n+13)E_1+E_T$	$13P$	DLIN
CLL+12 [6]	$(4n+3)\lvert G_1\rvert+\lvert G_T\rvert$	$(4n+3)E_2$	$(8n+6)E_1+E_T$	$(4n+3)P$	SXDH
Ours	$4\lvert G_1\rvert+\lvert G_T\rvert$	$2(n+1)E_2$	$2(n+1)E_1+E_T$	$4P$	SXDH
(Sec 7)	$6\lvert G_1\rvert+\lvert G_T\rvert$	$3(n+1)E_2$	$6(n+1)E_1+2E_T$	$6P$	DLIN
	$2(d+1)\lvert G_1\rvert+\lvert G_T\rvert$	$(d+1)(n+1)E_2$	$d(d+1)(n+1)E_1+dE_T$	$2(d+1)P$	d-LIN

Fig. 2. Comparison between existing and our HIBE schemes, where n is the depth parameter; in addition, E_N denotes G_N-exponentiation. In all of the prime-order constructions, $\lvert\mathsf{MPK}\rvert = \mathcal{O}(n\lvert G_1\rvert + n\lvert G_2\rvert + \lvert G_T\rvert)$ and $\lvert\mathsf{SK}\rvert = \mathcal{O}(n\lvert G_2\rvert)$. For T_{Dec}, we omitted the overhead of $\mathcal{O}(n)$ exponentiations associated with delegating a key before decrypting. Apart from [2], all of the schemes achieve full security.

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

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

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

The key novelty in the framework lies in identifying the role of **associativity**. Prior to this work, the general consensus is that instantiating the dual system encryption methodology requires some form of strong orthogonality, as indicated in the sequence of works on simulating properties of composite-order groups in the prime-order setting via the framework of dual pairing vector spaces [14, 15, 10]. In particular, constructions based on the latter framework implies that for all $(g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}$ and all

$(h_0, h_1, \ldots, h_n) \in \mathbb{H}^{n+1}$ drawn from the respective normal distributions, we have that for all $i = 1, \ldots, n$:

$$e(g_0, h_i) = e(g_i, h_0) = 1 \quad \text{and} \quad \forall j \neq i,\ e(g_j, h_i) = e(g_i, h_j) = 1$$

along with additional analogous requirements amongst the semi-functional components. We note that our framework does require an orthogonality property, but only in a weak sense.

HIBE from dual system groups. We construct a HIBE for depth n from dual system groups for parameter $n + 1$. The scheme is as follows, and shares a similar structure to those in [11, 2]:

$$\begin{aligned} \text{CT}_{\mathbf{x}} &:= \left(g_0,\ g_{n+1} g_1^{x_1} \cdots g_\ell^{x_\ell}, e(g_0, \text{MSK}) \cdot M \right). \\ \text{SK}_{\mathbf{y}} &:= \left(h_0,\ \text{MSK} \cdot (h_{n+1} h_1^{y_1} \cdots h_\ell^{y_\ell}),\ h_{\ell+1},\ \ldots,\ h_n \right). \end{aligned}$$

where ℓ is the length of both $\mathbf{x}$ and $\mathbf{y}$, MSK is uniformly sampled from $\mathbb{H}$, $(g_0, g_1, \ldots, g_{n+1}) \in \mathbb{G}^{n+2}$ and $(h_0, h_1, \ldots, h_{n+1}) \in \mathbb{H}^{n+2}$ are drawn from the respective normal distributions. Note that correctness follows from associativity. Our proof strategy relies on dual system encryption and follows that in [20], which in turn builds upon that in [11, 19]. We also extend the construction to obtain a compact spatial encryption scheme [1].

Simulating composite-order groups. We sketch our new tools for simulating composite-order groups in the prime-order setting, which is implicit in our instantiation of dual system groups. For our exposition, it is convenient to think of asymmetric composite-order groups $(\hat{G}, \hat{G}^*, \hat{G}_T)$ of order $N = p_1 p_2$ which is the product of two primes, endowed with an efficient bilinear map $\hat{e} : \hat{G} \times \hat{G}^* \to \hat{G}_T$. Let $\hat{g}_1, \hat{g}_2 \in \hat{G}$ denote random generators of $\hat{G}$ of orders p_1, p_2 respectively; define $\hat{g}_1^*, \hat{g}_2^* \in \hat{G}^*$ analogously. Observe that we have the following orthogonality property:

$$\hat{e}(\hat{g}_1, \hat{g}_2^*) = \hat{e}(\hat{g}_2, \hat{g}_1^*) = 1.$$

A useful property of composite-order groups, especially in the context of dual system encryption [11, 12], is that we can perform randomization by raising a group element to the power of a random exponent $a \leftarrow_{\text{R}} \mathbb{Z}_N$. This operation satisfy the following useful properties:

(hiding.) given $\hat{g}_1^a, (\hat{g}_1^*)^a$ along with $\hat{g}_1, \hat{g}_1^*, \hat{g}_2, \hat{g}_2^*$, the quantity $a \pmod{p_2}$ is completely hidden statistically;

(orthogonality.) for all a, we have $\hat{e}(\hat{g}_1^a, \hat{g}_2^*) = 1$.

(associativity.) for all $(\hat{h}, \hat{h}^*) \in \hat{G} \times \hat{G}^*$ and all $a \in \mathbb{Z}_N$, we have

$$\hat{e}(\hat{h}^a, \hat{h}^*) = \hat{e}(\hat{h}, (\hat{h}^*)^a)$$

Indeed, previous works [14, 15, 10, 6] showed how to achieve all three properties; moreover, the correctness of the ensuing IBE and functional encryption schemes relies on all three properties. The source of inefficiency in prior works comes from orthogonality: roughly speaking, if we want to perform ℓ independent randomizations in prime-order groups, then we require an ℓ-dimensional vector space. This means that to simulate a single group element in a composite-order group, we will need a ℓ-tuple from a prime-order group. We do not require orthogonality. We will construct our functional encryption schemes so that associativity suffices for correctness. A side-benefit is that the structure of our encryption schemes are even more similar to previous selectively-secure prime-order and fully-secure composite-order schemes.

Basic group structure. Following [14, 15, 10, 6], we will simulate $(\hat{G}, \hat{G}^*, \hat{G}_T)$ in a prime-order bilinear group (G_1, G_2, G_T) as follows: pick a random $\mathbf{B} \leftarrow_{\text{R}} \text{GL}_{d+1}(\mathbb{Z}_p)$ and define $\mathbf{B}^* := (\mathbf{B}^\top)^{-1}$ so that $\mathbf{B}^\top \mathbf{B}^*$

	$\lvert\mathsf{PP}\rvert$	$\lvert\mathsf{SP}\rvert$	$\mathbb{G}$	$\mathbb{H}$	$\mathbb{G}_T$	$\mathrm{ord}(h^*)$
Composite	$(n+3)\lvert G_N\rvert$	$(n+2)\lvert G_N\rvert$	G_N	G_N	G_T	p_2p_3
SXDH	$2(n+1)(\lvert G_1\rvert+\lvert G_2\rvert)$	$2(n+1)(\lvert G_1\rvert+\lvert G_2\rvert)$	G_1^2	G_2^2	G_T	p
DLIN	$6(n+1)(\lvert G_1\rvert+\lvert G_2\rvert)$	$3(n+1)(\lvert G_1\rvert+\lvert G_2\rvert)$	G_1^3	G_2^3	G_T	p
d-LIN	$d(d+1)(n+1)(\lvert G_1\rvert+\lvert G_2\rvert)$	$(d+1)(n+1)(\lvert G_1\rvert+\lvert G_2\rvert)$	G_1^{d+1}	G_2^{d+1}	G_T	p

Fig. 3. Parameters for dual system groups, where $N = p_1p_2p_3$ and p is the order of G_1 and G_2.

is the identity matrix. Consider the following map:

$$(\hat{G}, \hat{G}^*, \hat{G}_T) \mapsto (G_1^{d+1}, G_2^{d+1}, G_T)$$
$$(\hat{g}_1, \hat{g}_2, \hat{g}_1^*, \hat{g}_2^*) \mapsto (g_1^{\pi_L(\mathbf{B})}, g_1^{\pi_R(\mathbf{B})}, g_2^{\pi_L(\mathbf{B}^*)}, g_2^{\pi_R(\mathbf{B}^*)})$$

where g_1, g_2 are the respective generators for G_1 and G_2; π_L, π_R denote the projection maps that map a $(d+1) \times (d+1)$ matrix to the left d columns and right-most column respectively, along with the bilinear map $e : G_1^{d+1} \times G_2^{d+1} \to G_T$ given by $e(g_1^{\mathbf{x}}, g_2^{\mathbf{y}}) := e(g_1, g_2)^{\mathbf{x}^\top \mathbf{y}}$. Observe that we achieve orthogonality, namely:

$$e(g_1^{\pi_L(\mathbf{B})}, g_2^{\pi_R(\mathbf{B}^*)})^\top = e(g_1^{\pi_R(\mathbf{B})}, g_2^{\pi_L(\mathbf{B}^*)}) = (1, \ldots, 1)$$

Moreover, under the d-LIN assumption, the construction satisfies a computational subspace-hiding assumption analogous to the subgroup indistinguishability assumption in composite-order groups.

Randomizing group elements. We achieve randomization as follows: pick a random $\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}$ and replace $(\mathbf{B}, \mathbf{B}^*)$ with $(\mathbf{BA}, \mathbf{B}^*\mathbf{A}^\top)$. Observe that this transformation achieve the following properties:

(hiding.) given $g_1^{\pi_L(\mathbf{BA})}, g_2^{\pi_L(\mathbf{B}^*\mathbf{A}^\top)}$ along with $g_1, g_2, \mathbf{B}, \mathbf{B}^*$, the quantity $\mathbf{e}_{d+1}^\top \mathbf{A} \mathbf{e}_{d+1}$ is completely hidden statistically;

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

$$e(g_1^{\mathbf{BA}}, g_2^{\mathbf{B}^*}) = e(g_1^{\mathbf{B}}, g_2^{\mathbf{B}^*\mathbf{A}^\top}) = e(g_1, g_2)^{\mathbf{A}^\top}$$

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

Subsequent work. In [5], we presented the first adaptively secure IBE where the security loss does not depend on the number of secret key queries. We started with a construction in composite-order groups, and extended the techniques in this work to obtain an instantiation in prime-order groups. We also extended the dual system groups framework in this work to obtain a modular analysis.

Organization. We give the definition and security model of HIBE in Section 2. We present dual system groups in Section 3 and our HIBE scheme in Section 4. We present instantiations of dual system groups in Sections 5 and 6. We present a self-contained description of our HIBE scheme in Section 7.

2 Preliminaries

Notation. We denote by $s \leftarrow_R S$ the fact that s is picked uniformly at random from a finite set S and by $x, y, z \leftarrow_R S$ 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 $g \in G$.

Hierarchical Identity-Based Encryption. An HIBE scheme [8] consists of five algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec}, \mathsf{KeyDel})$:

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

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

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

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

$\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{y}') \to \mathrm{SK}_{\mathbf{y}'}$. The key delegation algorithm takes in the public parameters MPK, a secret key $\mathrm{SK}_{\mathbf{y}}$, and an identity vector $\mathbf{y}'$, where $\mathbf{y}$ is a prefix of $\mathbf{y}'$. It outputs a secret key $\mathrm{SK}_{\mathbf{y}'}$.

Correctness. For all $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, all identity vectors $\mathbf{x}$, all messages m, all decryption keys $\mathrm{SK}_{\mathbf{y}}$, all $\mathbf{x}$ such that $\mathbf{y}$ is a prefix of $\mathbf{x}$, we have

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

Delegation. We require that delegation is independent of the path taken; that is, if $\mathbf{y}$ is a prefix of $\mathbf{y}'$, then the following distributions are identical:

$$\{\mathrm{SK}_{\mathbf{y}}, \mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{y}')\} \quad \text{and} \quad \{\mathrm{SK}_{\mathbf{y}}, \mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y}')\}$$

Security Model. We now give the notation of *adaptive* security for HIBE. The security game is defined by the following experiment, played by a challenger and an adversary $\mathcal{A}$.

Challenge Space. The adversary $\mathcal{A}$ gives the challenger the depth parameter 1^n.

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

Phase 1. The adversary $\mathcal{A}$ adaptively requests keys for any identity vector $\mathbf{y}$ of its choice. The challenger responds with the corresponding secret key $\mathrm{SK}_{\mathbf{y}}$, which it generates by running $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$. Because of our restriction on delegation, the returned $\mathrm{SK}_{\mathbf{y}}$ is independent of the path taken.

Challenge. The adversary submits two messages m_0 and m_1 of equal length and a challenge identity vector $\mathbf{x}^*$ with the restriction that no queried identity vector in Phase 1 is a prefix of it. The challenger picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$, 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 as in Phase 1 with the restriction that any queried identity vector $\mathbf{y}$ must not be a prefix of $\mathbf{x}^*$.

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

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

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

3 Dual System Groups

3.1 Overview

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

(subgroup indistinguishability.) There are two computationally indistinguishable ways to sample correlated $(n+1)$-tuples from $\mathbb{G}^{n+1}$: the "normal" distribution, and a higher-entropy distribution with "semi-functional components". We sample the normal distribution using SampG and the semi-functional components using $\widehat{\mathsf{SampG}}$. An analogous property holds for $\mathbb{H}^{n+1}$, with algorithms SampH and $\widehat{\mathsf{SampH}}$ respectively, with an important distinction in the auxiliary input provided to the distinguisher. Note that we separately sample the normal distribution and semi-functional components, which makes parameter-hiding (defined in Section 3.2) easier to state.

(associativity.) For all $(g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}$ and all $(h_0, h_1, \ldots, h_n) \in \mathbb{H}^{n+1}$ drawn from the respective normal distributions according to SampG and SampH, 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.

(right subgroup $\mathbb{H}$.) There is some distinguished element $h^* \in \mathbb{H}$, which generates the semi-functional components in $\mathbb{H}$. It is convenient to think of h^* as being orthogonal to 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), so that we get a random value when we decrypt a semi-functional ciphertext with a semi-functional key.

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

Property	Where it is used	Remark
projective	correctness	
	Lemma 1	normal to semi-functional CT
associative	correctness	
orthogonality	Lemma 6	final transition
non-degeneracy	Lemma 4	pseudo-normal to pseudo-SF Keys
	Lemma 6	final transition
$\mathbb{H}$-subgroup	key delegation	
left subgroup indistinguishability	Lemma 1	normal to semi-functional CT
right subgroup indistinguishability	Lemma 2	normal to pseudo-normal keys
	Lemma 5	pseudo-SF to semi-functional keys
parameter-hiding	Lemma 4	pseudo-normal to pseudo-SF Keys

Fig. 4. Summary of dual system groups

random exponents $(\gamma_1, \ldots, \gamma_n)$ shared across $\mathbb{G}$ and $\mathbb{H}$. It is crucial here that we use the same γ_i in $\mathbb{G}$ and in $\mathbb{H}$, so that decryption succeeds with nominally semi-functional objects.

3.2 Definitions

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

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, output public and secret parameters $(\mathrm{PP}, \mathrm{SP})$, 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 $\hat{\mathbf{g}} \in \mathbb{G}^{n+1}$.

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

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

Correctness. The requirements for correctness are as follows:

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

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

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

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

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

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

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

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

where

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

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

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

where

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

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

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

where

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

Discussion. We provide a brief 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 groups in 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 groups in Section 6, h^* is in general not orthogonal to $g_1, \ldots, g_n$.

Remark 2 (non-degeneracy). We rely on non-degeneracy to information-theoretically hide the message in the final transition, where all the keys and ciphertexts are semi-functional.

Remark 3 ($\mathbb{H}$-subgroup). We rely on $\mathbb{H}$-subgroup to re-randomize secret keys in HIBE key delegation.

Remark 4 (indistinguishability). We stress that left subgroup and right subgroup indistinguishability are not symmetric (the distinguisher receives additional auxiliary input for the latter); this clarifies why the instantiation in symmetric composite-order groups uses two primes in the ciphertext space and three in the secret key space. In left subgroup indistinguishability, the distinguisher does not get h^*; otherwise, it is possible to distinguish between the two distributions using *orthogonality*. On the other hand, in right subgroup indistinguishability, the distinguisher does get h^*, along with a sample $\mathbf{g} \cdot \hat{\mathbf{g}}$ from the high entropy distribution over $\mathbb{G}^{n+1}$. In the proof, we use $\mathbf{g} \cdot \hat{\mathbf{g}}$ to compute the semi-functional challenge ciphertext, and h^* to sample MSK together with a semi-functional MSK in Lemma 2 and 5.

Remark 5 (associative). We may deduce the following "extended" associative relations from the basic associative property along with left and right subgroup indistinguishability:

$$e(\hat{g}_0, h_i) = e(\hat{g}_i, h_0) \quad \text{and} \quad e(g_0, \hat{h}_i) = e(g_i, \hat{h}_0) \quad \text{and} \quad e(\hat{g}_0, \hat{h}_i) = e(\hat{g}_i, \hat{h}_0).$$

for all

$$\begin{aligned} &(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathrm{PP}) \quad \text{and} \quad (\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_n) \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP}), \\ &(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathrm{PP}) \quad \text{and} \quad (\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_n) \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP}). \end{aligned}$$

More concretely, we have:

$$\begin{aligned} &(1) && e(g_0 \cdot \hat{g}_0, h_i) = e(g_i \cdot \hat{g}_i, h_0) && \text{(left subgroup indistinguishability and associative)} \\ &(2) && e(g_0, h_i \cdot \hat{h}_i) = e(g_i, h_0 \cdot \hat{h}_0) && \text{(right subgroup indistinguishability and associative)} \\ &(3) && (g_0 \cdot \hat{g}_0, h_i \cdot \hat{h}_i) = e(g_i \cdot \hat{g}_i, h_0 \cdot \hat{h}_0) && \text{(right subgroup indistinguishability and (1))} \end{aligned}$$

Combining the three equalities above together with the basic associative relation, we obtain the extended relations.

Remark 6 (parameter hiding). Here, γ_i corresponds to the value that is hidden in the public parameters, i.e. the information-theoretically hidden entropy in the semi-functional component, corresponding for instance, to the G_{p_2}-components in composite-order groups. It is crucial here that we have the same γ_i over $\mathbb{G}$ and over $\mathbb{H}$, which guarantees that a nominally-SF key can decrypt a semi-functional ciphertext.

Remark 7. Using the Lewko's framework [10]:

- if we generate n different constant-dimension bases, then we do not satisfy associative.
- if we generate a $O(n)$-dimensional basis, then each element in $\mathbb{G}$ or $\mathbb{H}$ has length $\Omega(n)$.

4 Compact HIBE from Dual System Groups

We provide a construction of a compact HIBE scheme from dual system groups where the ciphertext comprises two group elements in $\mathbb{G}$ and one in $\mathbb{G}_T$. The correctness of the scheme relies on generic properties of dual system groups; however, security requires an additional assumption, namely that $\mathrm{ord}(\mathbb{H})$ is prime, which is indeed satisfied by our instantiation of dual system groups in the prime-order setting. Later on, we describe how to relax this requirement (see Remark 11).

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 Boneh-Boyen-Goh HIBE [2] with hierarchical identity space $\mathbb{Z}^n_{\mathrm{ord}(\mathbb{H})}$:

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

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

4.1 Construction

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

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

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

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

$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m)$: On input an identity vector $\mathbf{x} := (x_1, \ldots, x_\ell) \in \mathbb{Z}^\ell_{\mathrm{ord}(\mathbb{H})}$ and $m \in \mathbb{G}_T$, sample

$$(g_0, g_1, \ldots, g_n, g_{n+1}) \leftarrow \mathsf{SampG}(\mathrm{PP}; s),\ g'_T \leftarrow \mathsf{SampGT}(\mu(\mathrm{MSK}); s)$$

and output

$$\mathrm{CT}_{\mathbf{x}} := \left(\ C_0 := g_0,\ C_1 := g_{n+1} \cdot g_1^{x_1} \cdots g_\ell^{x_\ell},\ C_2 := g'_T \cdot m\ \right) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T.$$

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$: On input an identity vector $\mathbf{y} := (y_1, \ldots, y_\ell) \in \mathbb{Z}^\ell_{\mathrm{ord}(\mathbb{H})}$, sample

$$(h_0, h_1, \ldots, h_n, h_{n+1}) \leftarrow \mathsf{SampH}(\mathrm{PP})$$

and output

$$\mathrm{SK}_{\mathbf{y}} := \left(\ K_0 := h_0,\ K_1 := \mathrm{MSK} \cdot h_{n+1} \cdot h_1^{y_1} \cdots h_\ell^{y_\ell},\ K_{\ell+1} := h_{\ell+1},\ \ldots,\ K_n := h_n\ \right) \in (\mathbb{H})^{n-\ell+2}.$$

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{y}$ is a prefix of $\mathbf{x}$, run

$$\mathrm{SK}_{\mathbf{x}} := (K_0, K_1, \ldots) \leftarrow \mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{x}).$$

Compute

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

and recover the message as

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

$\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{y}')$: On input a secret key $\mathrm{SK}_{\mathbf{y}} := (K_0, K_1, K_{\ell+1}, \ldots, K_n)$ and an identity vector $\mathbf{y}' := (y_1, \ldots, y_{\ell'}) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^{\ell'}$, compute

$$\widetilde{\mathrm{SK}}_{\mathbf{y}'} := \left(K_0,\ K_1 \cdot K_{\ell+1}^{y_{\ell+1}} \cdots K_{\ell'}^{y_{\ell'}},\ K_{\ell'+1},\ \ldots,\ K_n \right),$$

and sample $\mathrm{SK}' \leftarrow \mathsf{KeyGen}(\mathrm{MPK}, 1, \mathbf{y}')$. Output

$$\mathrm{SK}_{\mathbf{y}'} := \widetilde{\mathrm{SK}}_{\mathbf{y}'} \cdot \mathrm{SK}'$$

where $\cdot$ denotes entry-wise multiplication.

Delegation. Fix $\mathbf{y}$ and $\mathbf{y}'$ such that $\mathbf{y}$ is a prefix of $\mathbf{y}'$. Let $\widetilde{\mathrm{SK}}_{\mathbf{y}'}$ and SK' be the values computed by $\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{y}')$. It is easy to see that SK' lies in the support of $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y}')$. By linearity of KeyGen and the $\mathbb{H}$-subgroup property, multiplying by SK' re-randomizes the key and yields independence of the path taken (c.f. Section 2).

Correctness. It suffices to establish correctness for $\mathbf{x} = \mathbf{y}$ using the delegation property. Observe that for $\mathrm{CT}_{\mathbf{x}}, \mathrm{SK}_{\mathbf{x}}$,

$$\begin{aligned}
e(C_0, K_1)/e(C_1, K_0) &= e\Big(g_0, \mathrm{MSK} \cdot (h_{n+1} \cdot h_1^{x_1} \cdots h_\ell^{x_\ell})\Big) \cdot e\Big(g_{n+1} \cdot g_1^{x_1} \cdots g_\ell^{x_\ell}, h_0\Big)^{-1} \\
&= e(g_0, \mathrm{MSK}) \cdot \Big(e(g_0, h_{n+1}) \cdot e(g_0, h_1)^{x_1} \cdots e(g_0, h_\ell)^{x_\ell}\Big) \\
&\quad \cdot \Big(e(g_{n+1}, h_0) \cdot e(g_1, h_0)^{x_1} \cdots e(g_\ell, h_0)^{x_\ell}\Big)^{-1} \\
&= e(g_0, \mathrm{MSK})
\end{aligned}$$

where the last equality relies on *associative*, namely $e(g_0, h_i) = e(g_i, h_0)$ and $e(g_{n+1}, h_0) = e(g_0, h_{n+1})$. Finally, by *projective*, $g_T' = e(g_0, \mathrm{MSK})$. Correctness follows readily.

4.2 Proof of Security

We prove the following theorem:

Theorem 1. *Under the left and right subgroup indistinguishability (described in Section 3) and the additional requirement that* $\mathrm{ord}(\mathbb{H})$ *is prime, our HIBE scheme in Section 4.1 is adaptively secure (in the sense of Definition 1). More precisely, for any adversary* $\mathcal{A}$ *that makes at most* q *key queries against the HIBE scheme, there exist adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *such that:*

$$\mathsf{Adv}_{\mathcal{A}}(\lambda)^{\mathrm{HIBE}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathrm{LS}}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\mathrm{RS}}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{B}_3}^{\mathrm{RS}}(\lambda),$$

and

$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n),$$

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

The proof follows via a series of games, analogous to that in [20] (which are in turn different from that in [11]) and summarized in Fig. 5. To describe the games, we must first define semi-functional keys and ciphertexts. Following [20], 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 $\mathbf{x} := (x_1, \ldots, x_\ell) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^\ell$, $m \in \mathbb{G}_T$, and $\mathbf{t} := (T_0, T_1, \ldots, T_n, T_{n+1}) \in \mathbb{G}^{n+2}$, output

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

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

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

Auxiliary distributions.

Semi-functional master secret key.

$$\widehat{\mathrm{MSK}} := \mathrm{MSK} \cdot (h^*)^\alpha,$$

where $\boxed{\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}}$.

Semi-functional ciphertext.

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

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

$$\left(g_0 \cdot \hat{g}_0,\ (g_{n+1} \cdot \hat{g}_{n+1}) \cdot \prod_{k=1}^{\ell} (g_k \cdot \hat{g}_k)^{x_k},\ e(g_0 \cdot \hat{g}_0, \mathrm{MSK}) \cdot m \right),$$

where $(g_0, g_1, \ldots, g_n, g_{n+1}) \leftarrow \mathsf{SampG}(\mathrm{PP})$ and $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_n, \hat{g}_{n+1}) \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})$.

Pseudo-normal secret key.

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

where fresh $\boxed{\mathbf{h} \leftarrow \mathsf{SampH}(\mathrm{PP}) \text{ and } \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})}$ are chosen for each secret key; we can also write this distribution more explicitly as

$$\left(h_0 \cdot \hat{h}_0,\ \mathrm{MSK} \cdot (h_{n+1} \cdot \hat{h}_{n+1}) \cdot \prod_{k=1}^{\ell} (h_k \cdot \hat{h}_k)^{y_k},\ h_{\ell+1} \cdot \hat{h}_{\ell+1},\ \ldots,\ h_n \cdot \hat{h}_n \right)$$

where $(h_0, h_1, \ldots, h_n, h_{n+1}) \leftarrow \mathsf{SampH}(\mathrm{PP})$ and $(\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_n, \hat{h}_{n+1}) \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})$.

Game	Ciphertext $\mathrm{CT}_{\mathbf{x}^*}$	Secret Key $\mathrm{SK}_{\mathbf{y}}$
0 : real game	$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}^*, m_\beta)$ $(g_0, g_{n+1}\prod g_k^{x_k}, e(g_0, \mathrm{MSK}) \cdot m_\beta)$	$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$ $(h_0, \mathrm{MSK} \cdot h_{n+1}\prod h_k^{y_k}, \cdots)$
1 : semi-functional CT via left subgroup	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \boxed{\mathbf{g}\cdot\hat{\mathbf{g}}})$ $(g_0\hat{g}_0, (g_{n+1}\hat{g}_{n+1}) \cdot \prod(g_k\hat{g}_k)^{x_k}, e(g_0\hat{g}_0, \mathrm{MSK}) \cdot m_\beta)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h})$ $(—, —, —)$
2.i.1 : pseudo-normal SK via right subgroup	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \mathbf{g}\cdot\hat{\mathbf{g}})$ $(—, —, —)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \boxed{\mathbf{h}\cdot\hat{\mathbf{h}}})$ $(h_0\hat{h}_0, \mathrm{MSK} \cdot h_{n+1}\hat{h}_{n+1}\prod(h_k\hat{h}_k)^{y_k}, \cdots))$
2.i.2 : pseudo-SF SK via parameter-hiding	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \mathbf{g}\cdot\hat{\mathbf{g}})$ $(—, —, —)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\widehat{\mathrm{MSK}}}, \mathbf{y}; \mathbf{h}\cdot\hat{\mathbf{h}})$ $(—, \widehat{\mathrm{MSK}} \cdot h_{n+1}\hat{h}_{n+1}\prod(h_k\hat{h}_k)^{y_k}, —)$
2.i.3 : semi-functional SK via right group	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \mathbf{g}\cdot\hat{\mathbf{g}})$ $(—, —, —)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \widehat{\mathrm{MSK}}, \mathbf{y}; \boxed{\mathbf{h}})$ $(h_0, \widehat{\mathrm{MSK}} \cdot h_{n+1}\prod h_k^{y_k}, \cdots)$
3 : final game	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, \boxed{\mathtt{random}}; \mathrm{MSK}, \mathbf{g}\cdot\hat{\mathbf{g}})$ $(—, —, e(g_0\hat{g}_0, \mathrm{MSK}) \cdot \mathtt{random})$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \widehat{\mathrm{MSK}}, \mathbf{y}; \mathbf{h})$ $(—, —, —)$

Fig. 5. Sequence of games, where we drew a box to highlight the differences between each game and the preceding one, omitted the delegation terms in the explicit expression of $\mathrm{SK}_{\mathbf{y}}$, a dash (—) means the same as in the previous game, and Games 2.i.x refers to the i'th secret key. Here, the product Π denotes $\Pi_{k=1}^{\ell}$.

Pseudo-SF secret key.

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

where fresh $\mathbf{h} \leftarrow \mathsf{SampH}(\mathrm{PP})$ and $\hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})$ are chosen for each secret key; we can also write this distribution more explicitly as

$$\left(h_0 \cdot \hat{h}_0,\ \widehat{\mathrm{MSK}} \cdot (h_{n+1} \cdot \hat{h}_{n+1}) \cdot \prod_{k=1}^{\ell} (h_k \cdot \hat{h}_k)^{y_k},\ h_{\ell+1} \cdot \hat{h}_{\ell+1},\ \ldots,\ h_n \cdot \hat{h}_n \right)$$

where $(h_0, h_1, \ldots, h_n, h_{n+1}) \leftarrow \mathsf{SampH}(\mathrm{PP})$ and $(\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_n, \hat{h}_{n+1}) \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})$.

Semi-functional secret key.

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

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

$$\left(h_0,\ \widehat{\mathrm{MSK}} \cdot h_{n+1} \cdot \prod_{k=1}^{\ell} h_k^{y_k},\ h_{\ell+1},\ \ldots,\ h_n \right)$$

where $(h_0, h_1, \ldots, h_n, h_{n+1}) \leftarrow \mathsf{SampH}(\mathrm{PP})$. We note that the semi-functional key generation algorithm is identical to the normal key generation except that it replaces MSK with $\widehat{\mathrm{MSK}}$ as input.

Remark 8 (semi-functional keys). We note that semi-functional secret keys in our proof are the same as that in [20] and different from those in [11, 10, 16, 6]. Specifically, only the term associated with MSK of our semi-functional key has a SF-component (namely, the semi-functional key is identical to a normal key

except that MSK is replaced with $\widehat{\text{MSK}}$) whereas each term of a semi-functional key in [11, 10, 16, 6] has independently random SF-components.

Remark 9 (decryption capabilities). Fix identities $\mathbf{x}^*, \mathbf{y}$ such that $\mathbf{x}^*$ is a prefix of $\mathbf{y}$. Then,

- all types of secret key $\text{SK}_{\mathbf{y}}$ can decrypt a normal ciphertext $\text{CT}_{\mathbf{x}^*}$;
- a normal or pseudo-normal secret key $\text{SK}_{\mathbf{y}}$ can decrypt a semi-functional ciphertext $\text{CT}_{\mathbf{x}^*}$;
- when using a pseudo-SF or semi-functional secret key $\text{SK}_{\mathbf{y}}$ to decrypt a semi-functional ciphertext $\text{CT}_{\mathbf{x}^*}$, the message is masked by an additional term $e(\hat{g}_0, h^*)^{\alpha}$, which is non-zero with high probability.

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

- 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 semi-functional.
- $\mathsf{Game}_{2,i,1}$ for i from 1 to q, $\mathsf{Game}_{2,i,1}$ is the same as Game_1 except that the first $i-1$ keys are semi-functional, the last $q-i$ keys are normal while the i'th key is pseudo-normal.
- $\mathsf{Game}_{2,i,2}$ for i from 1 to q, $\mathsf{Game}_{2,i,2}$ is the same as Game_1 except that the first $i-1$ keys are semi-functional, the last $q-i$ keys are normal while the i'th key is pseudo-SF.
- $\mathsf{Game}_{2,i,3}$ for i from 1 to q, $\mathsf{Game}_{2,i,3}$ is the same as Game_1 except that the first i keys are semi-functional, the last $q-i$ keys are normal.
- Game_3: is the same as $\mathsf{Game}_{2,q,3}$, except that the challenge ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_T$.

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

Remark 10 (relation to functionality). The game sequence in our proof of security is fairly generic in the sense that it does not exploit the HIBE functionality except in the transition from pseudo-normal to pseudo-SF keys in Lemma 4. This means that when we extend our result to spatial encryption in Section 8, it suffices to just modify a single lemma.

4.3 Normal to Semi-Functional Ciphertext

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*

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

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

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

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

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

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

and proceeds as follows:

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

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

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

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

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

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

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

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

4.4 Normal to Pseudo-Normal Keys

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

$$|\mathsf{Adv}_{2,i-1,3}(\lambda) - \mathsf{Adv}_{2,i,1}(\lambda)| \leq \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_2}(\lambda),$$

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* $\mathsf{Time}(\mathcal{A})$*. (We note that* $\mathsf{Game}_{2,0,3}$ *is identical to* Game_1*.)*

Proof. The adversary $\mathcal{B}_2$ gets as input

$$(\mathrm{PP}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}, \mathbf{t}),$$

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

$$\mathbf{h} \leftarrow \mathsf{SampH}(\mathrm{PP}),\ \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP}),$$

and proceeds as follows:

Setup. Pick $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$ and set $\widehat{\mathrm{MSK}} := \mathrm{MSK} \cdot (h^*)^\alpha$. Output

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

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

$$\mathrm{SK}_{\mathbf{y}} \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \widehat{\mathrm{MSK}}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})) & \text{if } j < i \\ \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{t}) & \text{if } j = i \\ \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})) & \text{if } j > i \end{cases}.$$

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

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

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

Observe that when $\mathbf{t} = \mathbf{h}$, the output is identical to that in $\mathsf{Game}_{2,i-1,3}$; and when $\mathbf{t} = \mathbf{h} \cdot \hat{\mathbf{h}}$, the output is identical to that in $\mathsf{Game}_{2,i,1}$. We may therefore conclude that: $|\mathsf{Adv}_{2,i-1,3}(\lambda) - \mathsf{Adv}_{2,i,1}(\lambda)| \leq \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_2}(\lambda)$. □

4.5 Pseudo-Normal to Pseudo-SF Keys

First, we recall the following statistical lemma implicit in [11].

Lemma 3 (implicit in [11]). *For any prime p, for all $\mathbf{x} := (x_1, \ldots, x_{\ell^*}) \in \mathbb{Z}_p^{\ell^*}$ and $\mathbf{y} := (y_1, \ldots, y_\ell) \in \mathbb{Z}_p^{\ell}$, where $\mathbf{y}$ is not a prefix of $\mathbf{x}$, the following distribution is identically distributed to the uniform distribution over $\mathbb{Z}_p^{n-\ell+2}$:*

$$\{\gamma_1 x_1 + \cdots + \gamma_{\ell^*} x_{\ell^*} + \gamma_{n+1}, \gamma_1 y_1 + \cdots + \gamma_\ell y_\ell + \gamma_{n+1}, \gamma_{\ell+1}, \ldots, \gamma_n\},$$

where $\gamma_1, \ldots, \gamma_n, \gamma_{n+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p$.

Lemma 4 ($\mathsf{Game}_{2,i,1}$ to $\mathsf{Game}_{2,i,2}$). *For $i = 1, \ldots, q$, we have*

$$|\mathsf{Adv}_{2,i,1}(\lambda) - \mathsf{Adv}_{2,i,2}(\lambda)| = 0.$$

Proof. Observe that the only difference between $\mathsf{Game}_{2,i,1}$ and $\mathsf{Game}_{2,i,2}$ lies in that we replace MSK in $\mathsf{Game}_{2,i,1}$ with $\widehat{\mathrm{MSK}}$ in $\mathsf{Game}_{2,i,2}$ as input for the i'th secret key query, where $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$ and $\widehat{\mathrm{MSK}} := \mathrm{MSK} \cdot (h^*)^\alpha$. Thus, it suffices to establish the following:

> *Claim.* For all α, all $\mathbf{x} := (x_1, \ldots, x_{\ell^*}) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^{\ell^*}$ and $\mathbf{y} := (y_1, \ldots, y_\ell) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^{\ell}$, where $\mathbf{y}$ is not a prefix of $\mathbf{x}$, the following distributions are identically distributed:
>
> $$\{\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m_\beta; \mathrm{MSK}, \mathbf{g} \cdot \hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK}}, \mathbf{y}; \mathbf{h} \cdot \hat{\mathbf{h}})\} \quad \text{and}$$
> $$\{\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m_\beta; \mathrm{MSK}, \mathbf{g} \cdot \hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK} \cdot (h^*)^\alpha}, \mathbf{y}; \mathbf{h} \cdot \hat{\mathbf{h}})\}.$$

We defer the proof of the claim for now, and first explain how the lemma follows from the claim. Given $(\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha)$, we can output $\mathrm{MPK} := (\mathrm{PP}, \mu(\mathrm{MSK}))$ and generate the first $i-1$ semi-functional secret keys, and the remaining $q-i$ normal secret keys using

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot (h^*)^\alpha, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})) \quad \text{and} \quad \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP}))$$

respectively.

This would in turn imply that $\mathsf{Game}_{2,i,1}$ and $\mathsf{Game}_{2,i,2}$ are statistically indistinguishable. We note that this holds even if the adversary chooses $\mathbf{y}$ adaptively after seeing the challenge ciphertext $\mathrm{CT}_{\mathbf{x}^*}$, or if the challenge $\mathbf{x}^*$ is chosen after the adversary sees $\mathrm{SK}_{\mathbf{y}}$. □

Proof (of claim). By linearity, we have:

$$\begin{aligned}
\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m_\beta; \mathrm{MSK}, \mathbf{g} \cdot \hat{\mathbf{g}}) &= \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m_\beta; \mathrm{MSK}, \mathbf{g}) \cdot \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, 1; \mathrm{MSK}, \hat{\mathbf{g}}) \\
\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h} \cdot \hat{\mathbf{h}}) &= \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h}) \cdot \widehat{\mathsf{KeyGen}}(\mathrm{PP}, 1, \mathbf{y}; \hat{\mathbf{h}}) \\
\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot (h^*)^\alpha, \mathbf{y}; \mathbf{h} \cdot \hat{\mathbf{h}}) &= \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h}) \cdot \widehat{\mathsf{KeyGen}}(\mathrm{PP}, (h^*)^\alpha, \mathbf{y}; \hat{\mathbf{h}})
\end{aligned}$$

Therefore, it suffices to show that:

$$\{\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, 1; \mathrm{MSK}, \hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{1}, \mathbf{y}; \hat{\mathbf{h}})\} \quad \text{and}$$
$$\{\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, 1; \mathrm{MSK}, \hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{(h^*)^\alpha}, \mathbf{y}; \hat{\mathbf{h}})\}$$

are identically distributed. By parameter-hiding, we may replace $(\mathrm{PP}, h^*, \boxed{\hat{\mathbf{g}}, \hat{\mathbf{h}}})$ with $(\mathrm{PP}, h^*, \boxed{\hat{\mathbf{g}} \cdot \hat{\mathbf{g}}', \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}'})$, which means it suffices to show that:

$$\{\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, 1; \mathrm{MSK}, \hat{\mathbf{g}} \cdot \hat{\mathbf{g}}'), \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{1}, \mathbf{y}; \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}')\} \quad \text{and}$$
$$\{\mathrm{PP}, \mathrm{MSK}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, 1; \mathrm{MSK}, \hat{\mathbf{g}} \cdot \hat{\mathbf{g}}'), \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{(h^*)^\alpha}, \mathbf{y}; \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}')\}$$

are identically distributed. At this point, we expand the expressions for $\widehat{\mathsf{Enc}}$ and $\widehat{\mathsf{KeyGen}}$:

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, 1; \mathrm{MSK}, \hat{\mathbf{g}} \cdot \hat{\mathbf{g}}') = (\hat{g}_0, \hat{g}_{n+1} \cdot \hat{g}_1^{x_1} \cdots \hat{g}_{\ell^*}^{x_{\ell^*}} \cdot \hat{g}_0^{\gamma_{n+1} + x_1\gamma_1 + \cdots + x_{\ell^*}\gamma_{\ell^*}}, e(\hat{g}_0, \mathrm{MSK}))$$

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, 1, \mathbf{y}; \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}') = (\hat{h}_0, \hat{h}_{n+1} \cdot \hat{h}_1^{y_1} \cdots \hat{h}_\ell^{y_\ell} \cdot \hat{h}_0^{\gamma_{n+1} + y_1\gamma_1 + \cdots + y_\ell\gamma_\ell}, \hat{h}_{\ell+1} \cdot \hat{h}_0^{\gamma_{\ell+1}}, \ldots, \hat{h}_n \cdot \hat{h}_0^{\gamma_n})$$

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, (h^*)^\alpha, \mathbf{y}; \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}') = (\hat{h}_0, (h^*)^\alpha \cdot \hat{h}_{n+1} \cdot \hat{h}_1^{y_1} \cdots \hat{h}_\ell^{y_\ell} \cdot \hat{h}_0^{\gamma_{n+1} + y_1\gamma_1 + \cdots + y_\ell\gamma_\ell}, \hat{h}_{\ell+1} \cdot \hat{h}_0^{\gamma_{\ell+1}}, \ldots, \hat{h}_n \cdot \hat{h}_0^{\gamma_n})$$

Since h^* lies in the group generated by $\hat{h}_0$, we may replace $(h^*)^\alpha$ by $(\hat{h}_0)^{\alpha'}$ and "for all α" by "for all α'" and obtain a stronger claim. Now, by focusing on the exponents of the terms involving $\hat{g}_0$ and $\hat{h}_0$, it suffices to show that for all α':

$$\{\gamma_{n+1} + x_1\gamma_1 + \cdots + x_{\ell^*}\gamma_{\ell^*}, \gamma_{n+1} + y_1\gamma_1 + \cdots + y_\ell\gamma_\ell, \gamma_{\ell+1}, \ldots, \gamma_n\} \quad \text{and}$$
$$\{\gamma_{n+1} + x_1\gamma_1 + \cdots + x_{\ell^*}\gamma_{\ell^*}, \alpha' + \gamma_{n+1} + y_1\gamma_1 + \cdots + y_\ell\gamma_\ell, \gamma_{\ell+1}, \ldots, \gamma_n\}$$

are identically distributed. The last statement follows readily from Lemma 3. □

4.6 Pseudo-SF to Semi-Functional Keys

Lemma 5 ($\mathsf{Game}_{2,i,2}$ **to** $\mathsf{Game}_{2,i,3}$)**.** *For $i = 1, \ldots, q$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_3$ such that*

$$|\mathsf{Adv}_{2,i,2}(\lambda) - \mathsf{Adv}_{2,i,3}(\lambda)| \leq \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_3}(\lambda),$$

and $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The proof is completely analogous to Lemma 2, except we use $\widehat{\mathrm{MSK}}$ instead of MSK to generate the i'th key query. That is, $\mathcal{B}_3$ is exactly the same as $\mathcal{B}_2$, with the following change:

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

$$\mathrm{SK}_{\mathbf{y}} \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \widehat{\mathrm{MSK}}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})) & \text{if } j < i \\ \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\widehat{\mathrm{MSK}}}, \mathbf{y}; \mathbf{t}) & \text{if } j = i \\ \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})) & \text{if } j > i \end{cases}.$$

Observe that when $\mathbf{t} = \mathbf{h}$, the output is identical to that in $\mathsf{Game}_{2,i,3}$; and when $\mathbf{t} = \mathbf{h} \cdot \hat{\mathbf{h}}$, the output is identical to that in $\mathsf{Game}_{2,i,2}$. We may therefore conclude that: $|\mathsf{Adv}_{2,i,2}(\lambda) - \mathsf{Adv}_{2,i,3}(\lambda)| \leq \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_3}(\lambda)$. □

4.7 Final Transition

Lemma 6 ($\mathsf{Game}_{2,q,3}$ **to** Game_3)**.** *For any adversary $\mathcal{A}$, we have*

$$|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| = 0.$$

Proof. First, we sample $(\mathrm{MSK}, \widehat{\mathrm{MSK}})$ in both games as follows: pick $\widehat{\mathrm{MSK}} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$ and set $\mathrm{MSK} := \widehat{\mathrm{MSK}} \cdot (h^*)^\alpha$. We may then simulate key set-up and answer key queries given just $(\mathrm{PP}, \widehat{\mathrm{MSK}})$ as follows:

Setup. Observe that

$$\mu(\mathrm{MSK}) = \mu(\widehat{\mathrm{MSK}}) \cdot \mu((h^*)^\alpha) = \mu(\widehat{\mathrm{MSK}})$$

where in the last equality, we use *orthogonality* $\mu(h^*) = 1$. Output

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

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

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

Now, observe that the challenge ciphertext in $\mathsf{Game}_{2,q,3}$ is given by:

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

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:

$$C_2' = e(g_0 \cdot \hat{g}_0, \mathrm{MSK}) = e(g_0 \cdot \hat{g}_0, \widehat{\mathrm{MSK}} \cdot (h^*)^\alpha) = e(g_0 \cdot \hat{g}_0, \widehat{\mathrm{MSK}}) \cdot \boxed{e(\hat{g}_0, h^*)^\alpha},$$

where in the last equality, we use linearity and the fact that $e(g_0, (h^*)^\alpha) = 1$ (see Remark 1). Recall that $(\mathrm{PP}, \widehat{\mathrm{MSK}}, \mathbf{g} \cdot \hat{\mathbf{g}})$ are all statistically independent of $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$. Then, by *non-degeneracy*, given $\widehat{\mathrm{MSK}}$, all of the secret keys, along with (C_0, C_1) in the challenge ciphertext, the quantity

$$e(\hat{g}_0, h^*)^\alpha$$

is uniformly distributed over $\mathbb{G}_T$. This implies 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: $|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| = 0$. □

Remark 11. In our composite-order instantiation, we only have the weaker guarantee that $e(\hat{g}_0, h^*)^\alpha$ has at least 2λ bits of min-entropy, instead of being uniform over $\mathbb{G}_T$. We will modify the HIBE 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 $|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| \leq 2^{-\Omega(\lambda)}$. We will also require a composite-order analogue of Lemma 3 again implicit in [11], where we quantify over prefixes $\mathbf{x}, \mathbf{y}$ such that $\mathbf{y} \pmod{p_i}$ is not a prefix of $\mathbf{x} \pmod{p_i}$ for every prime divisor p_i of N. This restriction is essentially WLOG, since we may otherwise find a non-trivial factor of N from the adversary's key queries.

5 Instantiations in composite-order groups

In this section, we present an instantiation of dual system groups from subgroup decisional assumption in composite-order bilinear groups. The construction is implicit in [11].

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

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

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

$$(N, G_N, G_T, g_1, g_2, g_3, e) \leftarrow \mathcal{G}(1^\lambda);$$
$$h_{123} \leftarrow_{\mathrm{R}} G_N,\ h_{23} \leftarrow_{\mathrm{R}} G_{p_2 p_3},\ g_{12} \leftarrow_{\mathrm{R}} G_{p_1 p_2};$$
$$D := ((N, G_N, G_T, e); g_1, g_3, h_{123}, h_{23}, g_{12});$$
$$T_0 \leftarrow_{\mathrm{R}} G_{p_1 p_3},\ T_1 \leftarrow_{\mathrm{R}} G_N.$$

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);

Output

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

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 $(g_1^r \cdot g_3^r, g_1^{r\mathbf{w}} \cdot \mathbf{X}_3) \in G_{p_1 p_3}^{n+1}$.

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

$\widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})$: Pick $\hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^*$, $\mathbf{X}_3 \leftarrow_{\mathrm{R}} G_{p_3}^n$ and output $(g_2^{\hat{r}} \cdot g_3^{\hat{r}}, g_2^{\hat{r}\mathbf{w}} \cdot \mathbf{X}_3) \in G_{p_2 p_3}^{n+1}$.

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

$$(g_1^s, g_1^{sw_1}, \ldots, g_1^{sw_n}) \leftarrow \mathsf{SampG}(\mathrm{PP}) \quad \text{and} \quad (g_1^r \cdot g_3^r, g_1^{rw_1} \cdot X_{3,1}, \ldots, g_1^{rw_n} \cdot X_{3,n}) \leftarrow \mathsf{SampH}(\mathrm{PP})$$

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

$$e(g_1^s, g_1^{rw_i} \cdot X_{3,i}) = e(g_1, g_1)^{srw_i} = e(g_1^{sw_i}, g_1^r \cdot g_3^r).$$

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

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

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

where the final inequality follows from the fact that h^* is a generator of $G_{p_2p_3}$ and $\hat{s} \in \mathbb{Z}_N^*$; thus, $e(g_2^{\hat{s}}, h^*)^\alpha$ has at least $\log p_2$ bits of min-entropy, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. Moreover, for all $g_2^{\hat{r}} \cdot g_3^{\hat{r}} \leftarrow \widehat{\mathsf{SampH}}_0(\mathrm{PP}, \mathrm{SP}; \hat{r})$, $g_2^{\hat{r}} \cdot g_3^{\hat{r}}$ is a generator of $G_{p_2p_3}$ since $\hat{r} \in \mathbb{Z}_N^*$. Non-degeneracy for h^* follows readily.

We establish left subgroup indistinguishability, right subgroup indistinguishability, and parameter-hiding in next three subsections. The left and right subgroup indistinguishability relies on computational assumptions in composite-order groups, whereas parameter-hiding is unconditional.

5.3 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

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

Lemma 7 (DS1 to LS). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{LS}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{DS1}}_{\mathcal{B}}(\lambda) + 1/p_1 + 2/p_2 + 1/p_3.$$

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

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

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

where either $T \leftarrow_{\mathrm{R}} G_{p_1}$ or $T \leftarrow_{\mathrm{R}} G_{p_1p_2}$, and proceeds as follows:

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

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

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 $T \leftarrow_{\mathrm{R}} G_{p_1}$, the output is identical to $(\mathrm{PP}, \mathbf{g})$; and when $T \leftarrow_{\mathrm{R}} G_{p_1 p_2}$ and the G_{p_2}-component of T is not the identity, which occurs with probability at least $1 - 1/p_2$, the output is identical to $(\mathrm{PP}, \mathbf{g} \cdot \hat{\mathbf{g}})$. The lemma then follows readily. □

5.4 Right Subgroup Indistinguishability

We may rewrite the corresponding advantage function as:

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

where

$$\begin{aligned}
&(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\
&\mathbf{g} := (g_1^s, g_1^{s\mathbf{w}}),\ s \leftarrow_{\mathrm{R}} \mathbb{Z}_N;\\
&\hat{\mathbf{g}} := (g_2^{\hat{s}}, g_2^{\hat{s}\mathbf{w}}),\ \hat{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^*;\\
&\mathbf{h} := (g_1^r \cdot g_2^r, g_1^{r\mathbf{w}} \cdot \mathbf{X}_3),\ r \leftarrow_{\mathrm{R}} \mathbb{Z}_N,\ \mathbf{X}_3 \leftarrow_{\mathrm{R}} G_{p_3}^n;\\
&\hat{\mathbf{h}} := (g_2^{\hat{r}} \cdot g_3^{\hat{r}}, g_2^{\hat{r}\mathbf{w}} \cdot \mathbf{Y}_3),\ \hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^*,\ \mathbf{Y}_3 \leftarrow_{\mathrm{R}} G_{p_3}^n.
\end{aligned}$$

Lemma 8 (DS2 to RS). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{RS}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{DS2}}_{\mathcal{B}}(\lambda) + 1/p_2 + 4/p_2 + 2/p_3.$$

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

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

$$(\ (N, G_N, G_T, e); g_1, g_3, h_{123}, h_{23}, g_{12}, T\ ),$$

where either $T \leftarrow_{\mathrm{R}} G_{p_1 p_3}$ or $T \leftarrow_{\mathrm{R}} G_N$, and proceeds as follows:

Simulating auxiliary input $\mathrm{PP}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}$. Pick $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_N^n$ and output

$$\mathrm{PP} := (\ (N, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e);\ g_1, g_1^{\mathbf{w}}, g_3, h_{123}\ ) \quad \text{and} \quad h^* := h_{23} \quad \text{and} \quad \mathbf{g} \cdot \hat{\mathbf{g}} := (g_{12}, g_{12}^{\mathbf{w}}).$$

Observe that $\mathrm{PP}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}$ are properly distributed as long as h_{123} is a generator of G_N, h_{23} is a generator of $G_{p_2 p_3}$, and the G_{p_2}-component of g_{12} is not the identity; these occur with probability at least $1 - 1/p_1 - 3/p_2 - 2/p_3$.

Simulating the challenge. Pick $\mathbf{X}_3' \leftarrow_{\mathrm{R}} G_{p_3}^n$ and output $(T, T^{\mathbf{w}} \cdot \mathbf{X}_3')$.

Observe that when $T \leftarrow_{\mathrm{R}} G_{p_1 p_3}$, the output is identical to $(\mathrm{PP}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}, \mathbf{h})$; and when $T \leftarrow_{\mathrm{R}} G_N$ and the G_{p_2}-component of T is not the identity, which occurs with probability at least $1 - 1/p_2$, the output is identical to $(\mathrm{PP}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}, \mathbf{h} \cdot \hat{\mathbf{h}})$. The lemma then follows readily. □

5.5 Parameter-Hiding

We may rewrite the corresponding parameter-hiding as:

Lemma 9 (parameter-hiding). *The following distributions are identically distributed*

$$\left\{\ \mathrm{PP}, h^*, (g_2^{\hat{s}}, g_2^{\hat{s}\mathbf{w}}), (g_2^{\hat{r}} \cdot g_3^{\hat{r}}, g_2^{\hat{r}\mathbf{w}} \cdot \mathbf{X}_3)\ \right\} \quad \textit{and} \quad \left\{\ \mathrm{PP}, h^*, (g_2^{\hat{s}}, g_2^{\hat{s}(\mathbf{w}+\mathbf{w}')}), (g_2^{\hat{r}} \cdot g_3^{\hat{r}}, g_2^{\hat{r}(\mathbf{w}+\mathbf{w}')} \cdot \mathbf{X}_3)\ \right\}$$

where

$$(\text{PP}, \text{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\ \hat{s}, \hat{r} \leftarrow_{\text{R}} \mathbb{Z}_N^*;\ \mathbf{X}_3 \leftarrow_{\text{R}} G_{p_3}^n;\ \mathbf{w}' \leftarrow_{\text{R}} \mathbb{Z}_N^n.$$

Proof. Observe that h^* is a generator of $G_{p_2p_3}$ and PP has the form $((N, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); g_1, g_1^{\mathbf{w}}, g_3, h_{123})$, which depend only on $\mathbf{w}$ modulo p_1 since

$$g_1^{\mathbf{w}} = g_1^{\mathbf{w} \ (\bmod\ p_1\ )}.$$

The lemma follows readily from the Chinese Remainder Theorem, namely, $\mathbf{w}$ modulo p_1 and modulo p_2 are uncorrelated. □

6 Instantiations from d-LIN in prime-order groups

We provide an instantiation of dual system groups from d-LIN in asymmetric prime-order bilinear groups. The starting point of our construction uses ideas from dual vector pairing spaces [13, 14], but the final construction is fairly different.

Combined with our HIBE scheme in Section 4 and instantiation from d-LIN in Appendix 6, we obtain a depth n HIBE based on d-LIN with the following parameters:

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

A self-contained description of our HIBE scheme is given in Appendix 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:*

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

where

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

Remark 12. Typically, we sample $a_1, \ldots, a_d, s_{d+1} \leftarrow_{\text{R}} \mathbb{Z}_p$; this yields a $(d+1)/p$ negligible difference in the advantage.

Matrix-in-the-exponent. Given two vectors $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n)$ 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 $(g^{x_1}, \ldots, g^{x_n})$; 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 $(d+1) \times (d+1)$ matrix to the left d columns and right-most column 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 $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) := (G_1^{d+1}, G_2^{d+1}, G_T, e)$;
- sample $\mathbf{B} \leftarrow_{\mathrm{R}} \mathrm{GL}_{d+1}(\mathbb{Z}_p)$ and set $\mathbf{B}^* := (\mathbf{B}^{-1})^\top$; pick $\mathbf{A}_1, \ldots, \mathbf{A}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}$ and a random full-rank diagonal matrix $\mathbf{R}$ in $\mathbb{Z}_p^{(d+1)\times(d+1)}$ whose bottom-right entry is 1; define

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

- define $\mu : G_2^{d+1} \to G_T^d$ by $\mu(g_2^{\mathbf{k}}) = e(g_1^{\mathbf{D}}, g_2^{\mathbf{k}})$ for all $\mathbf{k} \in \mathbb{Z}_p^{d+1}$.
- Set $h^* := g_2^{\mathbf{f}^*}$;

Output

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

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 $g_T^{\mathbf{s}^\top\mathbf{p}} \in G_T$.

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

$\mathsf{SampH}(\mathrm{PP})$: Pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and output $(g_2^{\mathbf{D}^*\mathbf{r}}, g_2^{\mathbf{D}_1^*\mathbf{r}}, \ldots, g_2^{\mathbf{D}_n^*\mathbf{r}}) \in (G_2^{d+1})^{n+1}$.

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

$\widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})$: Pick $\hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and output $(g_2^{\hat{r}\mathbf{f}^*}, g_2^{\hat{r}\mathbf{f}_1^*}, \ldots, g_2^{\hat{r}\mathbf{f}_n^*}) \in (G_2^{d+1})^{n+1}$.

Correctness. We check correctness properties as follows:

(projective.) For all $\mathbf{k} \in \mathbb{Z}_p^{d+1}$ and all coin tosses $\mathbf{s} \in \mathbb{Z}_p^d$, we have $\mu(g_2^{\mathbf{k}}) = e(g_1, g_2)^{\mathbf{D}^\top\mathbf{k}}$ and

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

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

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

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

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

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

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

Security. We check security properties as follows:

(orthogonality.) For $g_1^{\mathbf{D}}$ and $g_2^{\mathbf{f}^*}$, we have

$$\mu(g_2^{\mathbf{f}^*}) = e(g_1^{\mathbf{D}}, g_2^{\mathbf{f}^*}) = (1, \ldots, 1)^\top,$$

where in the equality, we use the fact that

$$\mathbf{D}^\top\mathbf{f}^* = \pi_L(\mathbf{B})^\top\pi_R(\mathbf{B}^*\mathbf{R}) = \pi_L(\mathbf{B})^\top\pi_R(\mathbf{B}^*) = (0, \ldots, 0)^\top.$$

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

$$e(g_1^{\mathbf{D}\mathbf{s}} \cdot g_1^{\hat{s}\mathbf{f}}, g_2^{\mathbf{f}^*}) = e(g_1^{\hat{s}\mathbf{f}}, g_2^{\mathbf{f}^*}) = e(g_1, g_2)^{\hat{s}} \neq 1, \text{ since } \hat{s} \in \mathbb{Z}_p^*;$$

thus, $e(g_1^{\mathbf{D}\mathbf{s}} \cdot g_1^{\hat{s}\mathbf{f}}, g_2^{\mathbf{f}^*})^\alpha$ is identically distributed to the uniform distribution over G_T, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$. Moreover, for all $g_2^{\hat{r}\mathbf{f}^*} \leftarrow \widehat{\mathsf{SampH}}_0(\mathrm{PP}, \mathrm{SP}; \hat{r})$, it is clear that $g_2^{\mathbf{f}^*}$ lies in the group generated by $g_2^{\hat{r}\mathbf{f}^*}$ since $\hat{r} \in \mathbb{Z}_p^*$.

We establish left subgroup indistinguishability, right subgroup indistinguishability, and parameter-hiding in next three subsections. The left and right subgroup indistinguishability relies on the d-LIN assumption in prime-order groups, whereas parameter-hiding is unconditional.

6.3 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

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

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

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

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:

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

and

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

where $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and $\hat{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ (and thus $\binom{\mathbf{s}}{0}, \binom{\mathbf{s}}{\hat{s}} \in \mathbb{Z}_p^{d+1}$).
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 $\mathbf{s}, \hat{s}$. $\mathcal{B}$ implicitly sets

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

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

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

$$\mathbf{W} := \begin{pmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_d \\ a_{d+1} & a_{d+1} & \cdots & a_{d+1} & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{W}^* := \begin{pmatrix} a_1^{-1} & & & -a_1^{-1}a_{d+1} \\ & a_2^{-1} & & -a_2^{-1}a_{d+1} \\ & & \ddots & \vdots \\ & & a_d^{-1} & -a_d^{-1}a_{d+1} \\ & & & 1 \end{pmatrix}.$$

Observe that $\mathbf{W}^\top \mathbf{W}^* = \mathbf{I}_{d+1}$. It follows immediately that

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

Next, $\mathcal{B}$ samples $\tilde{\mathbf{B}} \leftarrow_{\mathrm{R}} \mathrm{GL}_{d+1}(\mathbb{Z}_p)$, along with $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}$, and implicitly sets:

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

It is clear that $(\mathbf{B}, \mathbf{B}^*)$ and $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are properly distributed. Note that we have:

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

Finally, $\mathcal{B}$ picks $\tilde{r}_1, \ldots, \tilde{r}_d \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and implicitly sets

$$\mathbf{R} := \begin{pmatrix} a_1\tilde{r}_1 & & & & \\ & a_2\tilde{r}_2 & & & \\ & & \ddots & & \\ & & & a_d\tilde{r}_d & \\ & & & & 1 \end{pmatrix}$$

Observe that $\mathbf{R}$ is properly distributed since $a_1, \ldots, a_d \neq 0$.

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

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

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 $\pi_L(\mathbf{W}^*\mathbf{R})$. Here, we use the fact that

$$\pi_L(\mathbf{W}^*\mathbf{R}) = \begin{pmatrix} \tilde{r}_1 & & & \\ & \tilde{r}_2 & & \\ & & \ddots & \\ & & & \tilde{r}_d \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

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

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

along with

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

Observe that if $\hat{s} = s_{d+1} = 0$, then the output challenge equals $\mathbf{g}$ and if $\hat{s} = s_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, then the output challenge equals $\mathbf{g} \cdot \hat{\mathbf{g}}$.

The lemma then follows readily. □

6.4 Right Subgroup Indistinguishability

We may rewrite the corresponding advantage function as:

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

where

$$\begin{aligned}
&(\text{PP}, \text{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\
&\mathbf{g} := (g_1^{\mathbf{D}\mathbf{s}}, g_1^{\mathbf{D}_1\mathbf{s}}, \ldots, g_1^{\mathbf{D}_n\mathbf{s}}),\ \mathbf{s} \leftarrow_{\text{R}} \mathbb{Z}_p^d;\\
&\hat{\mathbf{g}} := (g_1^{\hat{s}\mathbf{f}}, g_1^{\hat{s}\mathbf{f}_1}, \ldots, g_1^{\hat{s}\mathbf{f}_n}),\ \hat{s} \leftarrow_{\text{R}} \mathbb{Z}_p^*;\\
&\mathbf{h} := (g_2^{\mathbf{D}^*\mathbf{r}}, g_2^{\mathbf{D}_1^*\mathbf{r}}, \ldots, g_2^{\mathbf{D}_n^*\mathbf{r}}),\ \mathbf{r} \leftarrow_{\text{R}} \mathbb{Z}_p^d;\\
&\hat{\mathbf{h}} := (g_2^{\hat{r}\mathbf{f}^*}, g_2^{\hat{r}\mathbf{f}_1^*}, \ldots, g_2^{\hat{r}\mathbf{f}_n^*}),\ \hat{r} \leftarrow_{\text{R}} \mathbb{Z}_p^*.
\end{aligned}$$

Lemma 11 (**d-LIN to RS).** *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\text{RS}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{d\text{-LIN}}_{\mathcal{B}}(\lambda) + 1/p.$$

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 $(\text{PP}, h^*, \mathbf{g}\cdot\hat{\mathbf{g}}, \mathbf{h}, \mathbf{h}\cdot\hat{\mathbf{h}})$ in term of $\mathbf{B}, \mathbf{B}^*, \mathbf{A}_1, \ldots, \mathbf{A}_n, \mathbf{R}$ as follows:

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

and

$$\begin{aligned}
\mathbf{g} &:= (g_1^{\mathbf{B}\binom{\mathbf{s}}{0}}, g_1^{\mathbf{B}\mathbf{A}_1\binom{\mathbf{s}}{0}}, \ldots, g_1^{\mathbf{B}\mathbf{A}_n\binom{\mathbf{s}}{0}}),\\
\mathbf{g}\cdot\hat{\mathbf{g}} &:= (g_1^{\mathbf{B}\binom{\mathbf{s}}{\hat{s}}}, g_1^{\mathbf{B}\mathbf{A}_1\binom{\mathbf{s}}{\hat{s}}}, \ldots, g_1^{\mathbf{B}\mathbf{A}_n\binom{\mathbf{s}}{\hat{s}}}),\\
\mathbf{h} &:= (g_2^{\mathbf{B}^*\mathbf{R}\binom{\mathbf{r}}{0}}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}\binom{\mathbf{r}}{0}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R}\binom{\mathbf{r}}{0}}),\\
\mathbf{h}\cdot\hat{\mathbf{h}} &:= (g_2^{\mathbf{B}^*\mathbf{R}\binom{\mathbf{r}}{\hat{r}}}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R}\binom{\mathbf{r}}{\hat{r}}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R}\binom{\mathbf{r}}{\hat{r}}}),
\end{aligned}$$

where $\mathbf{s}, \mathbf{r} \leftarrow_{\text{R}} \mathbb{Z}_p^d$ and $\hat{s}, \hat{r} \leftarrow_{\text{R}} \mathbb{Z}_p^*$ (and thus $\binom{\mathbf{s}}{0}, \binom{\mathbf{s}}{\hat{s}}\binom{\mathbf{r}}{0}, \binom{\mathbf{r}}{\hat{r}} \in \mathbb{Z}_p^{d+1}$).
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 $r_{d+1} \leftarrow_{\text{R}} \mathbb{Z}_p^*$, and proceeds as follows:

Programming $\mathbf{r}, \hat{r}, \mathbf{R}$. $\mathcal{B}$ picks $\tilde{r}_1, \ldots, \tilde{r}_d \leftarrow_{\text{R}} \mathbb{Z}_p^*$ and implicitly sets

$$\mathbf{R} := \begin{pmatrix} a_1\tilde{r}_1 & & & & \\ & a_2\tilde{r}_2 & & & \\ & & \ddots & & \\ & & & a_d\tilde{r}_d & \\ & & & & 1 \end{pmatrix}$$

and

$$\mathbf{r} := (\tilde{r}_1^{-1} r_1, \ldots, \tilde{r}_d^{-1} r_d)^\top \quad \text{and} \quad \hat{r} := r_{d+1},$$

where $(r_1, \ldots, r_d, r_{d+1})$ are as defined in its input. Later on, $\mathcal{B}$ will output $\mathbf{h}$ if $\hat{r} = r_{d+1} = 0$, and $\mathbf{h} \cdot \hat{\mathbf{h}}$ if $\hat{r} = r_{d+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$. Observe that $\mathbf{R}$, $\mathbf{r}$, and $\hat{r}$ are independent and properly distributed as long as $a_1, \ldots, a_d \neq 0$.

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

$$\mathbf{W} := \begin{pmatrix} 1 & & & -a_1^{-1} a_{d+1} \\ & 1 & & -a_2^{-1} a_{d+1} \\ & & \ddots & \vdots \\ & & & 1 \quad -a_d^{-1} a_{d+1} \\ & & & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{W}^* := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_1^{-1} a_{d+1} & a_2^{-1} a_{d+1} & \cdots & a_d^{-1} a_{d+1} \; 1 \end{pmatrix},$$

Observe that $\mathbf{W}^\top \mathbf{W}^* = \mathbf{I}_{d+1}$. It follows immediately that

$$g_2^{\mathbf{W}^* \mathbf{R} \binom{\mathbf{r}}{\hat{r}}} = \begin{pmatrix} g_2^{a_1 r_1} \\ \vdots \\ g_2^{a_d r_d} \\ g_2^{a_{d+1}(r_1 + \cdots + r_d) + r_{d+1}} \end{pmatrix}$$

Next, $\mathcal{B}$ samples $\tilde{\mathbf{B}} \leftarrow_{\mathrm{R}} \mathrm{GL}_{d+1}(\mathbb{Z}_p)$, along with $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}$, and implicitly sets:

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

It is clear that $(\mathbf{B}, \mathbf{B}^*)$ and $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are properly distributed. Note that we have:

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

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

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

since it knows $(\tilde{\mathbf{B}}, \tilde{\mathbf{B}}^*, \tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_n), \pi_L(\mathbf{W})$ as well as $g_2^{\mathbf{W}^*\mathbf{R}}$. Here, we use the fact that

$$g_2^{\mathbf{W}^*\mathbf{R}} = \begin{pmatrix} (g_2^{a_1})^{\tilde{r}_1} & & & & \\ & (g_2^{a_2})^{\tilde{r}_2} & & & \\ & & \ddots & & \\ & & & (g_2^{a_d})^{\tilde{r}_d} & \\ (g_2^{a_{d+1}})^{\tilde{r}_1} & (g_2^{a_{d+1}})^{\tilde{r}_2} & \cdots & (g_2^{a_{d+1}})^{\tilde{r}_d} & g_2 \end{pmatrix}.$$

Simulating h^*. Observe that $\mathcal{B}$ can compute

$$h^* = g_2^{\pi_R(\mathbf{B}^*\mathbf{R})} = g_2^{\pi_R(\tilde{\mathbf{B}}^*\mathbf{W}^*\mathbf{R})},$$

since it knows $\tilde{\mathbf{B}}^*$ as well as $g_2^{\mathbf{W}^*\mathbf{R}}$.

Simulating auxiliary input $\mathbf{g}\cdot\hat{\mathbf{g}}$. $\mathcal{B}$ picks $\tilde{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{d+1}$ and implicitly sets

$$\begin{pmatrix}\mathbf{s}\\ \hat{s}\end{pmatrix} := \mathbf{W}^{-1}\tilde{\mathbf{s}}$$

Observe that $\mathbf{s}$ and $\hat{s}$ are properly distributed as long as $\hat{s}\neq 0$, which occurs with probability $1-1/p$. Now, we can write $\mathbf{g}\cdot\hat{\mathbf{g}}$ as:

$$g_1^{\mathbf{B}\binom{\mathbf{s}}{\hat{s}}} = g_1^{(\tilde{\mathbf{B}}\mathbf{W})(\mathbf{W}^{-1}\tilde{\mathbf{s}})} = g_1^{\tilde{\mathbf{B}}\tilde{\mathbf{s}}} \quad\text{and}\quad g_1^{\mathbf{B}\mathbf{A}_i\binom{\mathbf{s}}{\hat{s}}} = g_1^{(\tilde{\mathbf{B}}\tilde{\mathbf{A}}_i\mathbf{W})(\mathbf{W}^{-1}\tilde{\mathbf{s}})} = g_1^{\tilde{\mathbf{B}}\tilde{\mathbf{A}}_i\tilde{\mathbf{s}}},\ i=1,\ldots,n$$

where $(\tilde{\mathbf{B}},\tilde{\mathbf{A}}_1,\ldots,\tilde{\mathbf{A}}_n,\tilde{\mathbf{s}})$ is known to $\mathcal{B}$. Therefore, $\mathcal{B}$ can simulate auxiliary input $\mathbf{g}\cdot\hat{\mathbf{g}}$.

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

$$g_2^{\mathbf{B}^*\mathbf{R}\binom{\mathbf{r}}{\hat{r}}} = g_2^{\tilde{\mathbf{B}}^*\mathbf{W}^*\mathbf{R}\binom{\mathbf{r}}{\hat{r}}} = g_2^{\tilde{\mathbf{B}}^*\begin{pmatrix} a_1r_1\\ \vdots\\ a_dr_d\\ a_{d+1}(r_1+\cdots+r_d)+r_{d+1}\end{pmatrix}}$$

along with

$$g_1^{\mathbf{B}^*\mathbf{A}_i^\top\mathbf{R}\binom{\mathbf{r}}{\hat{r}}} = g_1^{\tilde{\mathbf{B}}^*\tilde{\mathbf{A}}_i^\top\mathbf{W}^*\mathbf{R}\binom{\mathbf{r}}{\hat{r}}} = g_2^{\tilde{\mathbf{B}}^*\tilde{\mathbf{A}}_i^\top\begin{pmatrix} a_1r_1\\ \vdots\\ a_dr_d\\ a_{d+1}(r_1+\cdots+r_d)+r_{d+1}\end{pmatrix}} \qquad i=1,\ldots,n$$

Observe that if $\hat{r}=r_{d+1}=0$, then the output challenge equals $\mathbf{h}$ and if $\hat{r}=r_{d+1}\leftarrow_{\mathrm{R}}\mathbb{Z}_p^*$, then the output challenge equals $\mathbf{h}\cdot\hat{\mathbf{h}}$.

The lemma then follows readily. □

6.5 Parameter-Hiding

We may rewrite the corresponding parameter-hiding as:

Lemma 12 (parameter-hiding). *The following distributions are identically distributed*

$$\left\{\mathrm{PP}, g_2^{\mathbf{f}^*}, \begin{matrix} g_1^{\hat{s}\mathbf{f}}, & g_1^{\hat{s}\mathbf{f}_1}, & \ldots, & g_1^{\hat{s}\mathbf{f}_n}\\ g_2^{\hat{r}\mathbf{f}^*}, & g_2^{\hat{r}\mathbf{f}_1^*}, & \ldots, & g_2^{\hat{r}\mathbf{f}_n^*}\end{matrix}\right\} \quad \textit{and} \quad \left\{\mathrm{PP}, g_2^{\mathbf{f}^*}, \begin{matrix} g_1^{\hat{s}\mathbf{f}}, & g_1^{\hat{s}(\mathbf{f}_1+\gamma_1\mathbf{f})}, & \ldots, & g_1^{\hat{s}(\mathbf{f}_n+\gamma_n\mathbf{f})}\\ g_2^{\hat{r}\mathbf{f}^*}, & g_2^{\hat{r}(\mathbf{f}_1^*+\gamma_1\mathbf{f}^*)}, & \ldots, & g_2^{\hat{r}(\mathbf{f}_n^*+\gamma_n\mathbf{f}^*)}\end{matrix}\right\}$$

where $(\mathrm{PP},\mathrm{SP})\leftarrow\mathsf{SampP}(1^\lambda,1^n)$, $\hat{s},\hat{r}\leftarrow_{\mathrm{R}}\mathbb{Z}_p^*$ *and* $\gamma_1,\ldots,\gamma_n\leftarrow_{\mathrm{R}}\mathbb{Z}_p$.

Proof. Fix $g_1, g_2, (\mathbf{B}, \mathbf{B}^*)$ and $\mathbf{R}$ (that is, we prove that the statement holds for all $g_1, g_2, \mathbf{B}, \mathbf{B}^*, \mathbf{R}$). Following $\mathsf{SampP}(1^\lambda, 1^n)$, we sample random $\mathbf{A}_1, \ldots, \mathbf{A}_n$, set

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

Next, let $\mathbf{V} := \mathbf{e}_{d+1}\mathbf{e}_{d+1}^\top \in \mathbb{Z}_p^{(d+1)\times(d+1)}$, that is $\mathbf{V}$ is the zero matrix with the bottom-right entry replaced with 1. Define matrices

$$\mathbf{A}'_i := \mathbf{A}_i + \gamma_i \mathbf{V}, \ i = 1, \ldots, n.$$

Now, consider the following probability experiment: we run SampP with $(\mathbf{A}'_1, \ldots, \mathbf{A}'_n)$ in place of $(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ to generate $(\mathrm{PP}, \mathrm{SP})$ and output

$$\{\mathrm{PP}, h^*, \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP}), \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP})\}$$

Observe that

$$\mathbf{B}\mathbf{A}'_i = \mathbf{B}\mathbf{A}_i + \gamma_i \mathbf{B}\mathbf{V} = [\ \mathbf{D}_i \parallel \mathbf{f}_i + \gamma_i \mathbf{f}\ ]$$

where in the second equality, we use the facts that (1) the maps π_L and π_R are linear; and (2) $\mathbf{B}\mathbf{V} = \mathbf{f}\mathbf{e}_{d+1}^\top$ and thus $\pi_R(\mathbf{B}\mathbf{V}) = \mathbf{f}$. Similarly,

$$\mathbf{B}^*(\mathbf{A}'_i)^\top\mathbf{R} = \mathbf{B}^*\mathbf{A}_i\mathbf{R} + \gamma_i \mathbf{B}^*\mathbf{V}^\top\mathbf{R} = [\ \mathbf{D}_i^* \parallel \mathbf{f}_i^* + \gamma_i \mathbf{f}^*\ ]$$

Here, we also use the fact that $\mathbf{V}^\top\mathbf{R} = \mathbf{V}$, since $\mathbf{R}$ is a diagonal matrix with bottom-right entry 1. Observe that:

- if $\gamma_1 = \cdots = \gamma_n = 0$, then we obtain the left distribution in the statement of the lemma;
- if $\gamma_1, \ldots, \gamma_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, then we obtain the right distribution in the statement of the lemma;
- whether we use $\gamma_1 = \cdots = \gamma_n = 0$ or $\gamma_1, \ldots, \gamma_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, we exactly obtain the same distribution for $\mathbf{A}_1, \ldots, \mathbf{A}_n$ and $\mathbf{A}'_1, \ldots, \mathbf{A}'_n$.

The lemma follows readily from combining the three observations. □

7 Concrete HIBE scheme from d-LIN in prime-order groups

In this section, we show how the concrete HIBE scheme from d-LIN works in prime-order groups. Recall that $\pi_L : \mathbb{Z}_p^{(d+1)\times(d+1)} \to \mathbb{Z}_p^{(d+1)\times d}$ is the projection map that maps a $(d+1)\times(d+1)$ matrix to the left d columns.

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

$$\mathbf{B}, \mathbf{B}^*, \mathbf{R} \leftarrow_{\mathrm{R}} \mathrm{GL}_{d+1}(\mathbb{Z}_p), \ \mathbf{A}_1, \ldots, \mathbf{A}_{n+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}, \ \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{d+1}$$

such that $\mathbf{B}^\top\mathbf{B}^* = \mathbf{I}$ and $\mathbf{R}$ is a diagonal matrix whose bottom-right entry is 1, and output the master public and secret key pair

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

and

$$\mathrm{MSK} := g_2^{\mathbf{k}} \in G_2^{d+1}.$$

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

$$\mathrm{CT}_{\mathbf{x}} := \left(C_0 := g_1^{\pi_L(\mathbf{B})\mathbf{s}},\ C_1 := g_1^{\pi_L(\mathbf{B}(\mathbf{A}_{n+1}+x_1\mathbf{A}_1+\cdots+x_\ell\mathbf{A}_\ell))\mathbf{s}},\ C_2 := e(g_1,g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \cdot m \right) \in G_1^{d+1} \times G_1^{d+1} \times G_T.$$

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

$$\mathrm{SK}_{\mathbf{y}} := \begin{pmatrix} K_0 := g_2^{\pi_L(\mathbf{B}^*\mathbf{R})\mathbf{r}},\ K_1 := g_2^{\mathbf{k}+\pi_L(\mathbf{B}^*(\mathbf{A}_{n+1}+y_1\mathbf{A}_1+\cdots+y_\ell\mathbf{A}_\ell)^\top\mathbf{R})\mathbf{r}} \\ K_{\ell+1} := g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_{\ell+1}^\top\mathbf{R})\mathbf{r}},\ \ldots,\ K_n := g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R})\mathbf{r}} \end{pmatrix} \in (G_2^{d+1})^{n-\ell+2}.$$

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{y}$ is a prefix of $\mathbf{x}$, run

$$\mathrm{SK}_{\mathbf{x}} := (K_0, K_1, \ldots) \leftarrow \mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{x}).$$

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

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

$\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{y}')$: On input a secret key $\mathrm{SK}_{\mathbf{y}} := (K_0, K_1, K_{\ell+1}, \ldots, K_n)$ and an identity vector $\mathbf{y}' := (y_1, \ldots, y_{\ell'}) \in \mathbb{Z}_p^{\ell'}$, first compute

$$\widetilde{\mathrm{SK}}_{\mathbf{y}'} := \left(K_0,\ K_1 \cdot K_{\ell+1}^{y_{\ell+1}} \cdots K_{\ell'}^{y_{\ell'}},\ K_{\ell'+1},\ \ldots,\ K_n \right).$$

Then, pick $\mathbf{r}' \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and compute

$$\mathrm{SK}' := \left(g_2^{\pi_L(\mathbf{B}^*\mathbf{R})\mathbf{r}'},\ g_2^{\pi_L(\mathbf{B}^*(\mathbf{A}_{n+1}+y_1\mathbf{A}_1+\cdots+y_{\ell'}\mathbf{A}_{\ell'})^\top\mathbf{R})\mathbf{r}'},\ g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_{\ell'+1}^\top\mathbf{R})\mathbf{r}'},\ \ldots,\ g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R})\mathbf{r}'} \right).$$

Finally, output

$$\mathrm{SK}_{\mathbf{y}'} := \widetilde{\mathrm{SK}}_{\mathbf{y}'} \cdot \mathrm{SK}'$$

where $\cdot$ denotes entry-wise multiplication.

8 Spatial Encryption from Dual System Groups

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

8.1 Spatial Encryption

For any matrix $\mathbf{Y} \in \mathbb{Z}_p^{n\times\ell}$, we use $\mathsf{span}(\mathbf{Y})$ to denote the linear space spanned by columns of $\mathbf{Y}$. A spatial encryption scheme consists of five algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec}, \mathsf{KeyDel})$:

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

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

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

$\mathsf{Dec}(\text{MPK}, \text{SK}_{\mathbf{Y}}, \text{CT}_{\mathbf{x}}) \to m$. The decryption algorithm takes in the public parameters MPK, a secret key $\text{SK}_{\mathbf{Y}}$ for a space $\mathbf{Y}$, and a ciphertext $\text{CT}_{\mathbf{x}}$ encrypted under a vector $\mathbf{x}$. It outputs a message m if $\mathbf{x} \in \mathsf{span}(\mathbf{Y})$.

$\mathsf{KeyDel}(\text{MPK}, \text{SK}_{\mathbf{Y}}, \mathbf{Y}') \to \text{SK}_{\mathbf{Y}'}$. The key delegation algorithm takes in the public parameters MPK, a secret key $\text{SK}_{\mathbf{Y}}$, and a space $\mathbf{Y}'$, where $\mathsf{span}(\mathbf{Y}') \subseteq \mathsf{span}(\mathbf{Y})$. It outputs a secret key $\text{SK}_{\mathbf{Y}'}$.

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

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

Delegation. We require that delegation is independent of the path taken; that is, if $\mathbf{Y}' \subseteq \mathbf{Y}$, then the following distributions are identical:

$$\{\text{SK}_{\mathbf{Y}}, \mathsf{KeyDel}(\text{MPK}, \text{SK}_{\mathbf{Y}}, \mathbf{Y}')\} \quad \text{and} \quad \{\text{SK}_{\mathbf{Y}}, \mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{Y}')\}$$

8.2 Security Model

We now give the notation of *adaptive* security for spatial encryption. The security game is defined by the following experiment, played by a challenger and an adversary $\mathcal{A}$.

Challenge Space. The adversary $\mathcal{A}$ gives the challenger the dimension parameter 1^n.

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 space $\mathbf{Y}$ of its choice. The challenger responds with the corresponding secret key $\text{SK}_{\mathbf{Y}}$, which it generates by running $\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{Y})$. Because of our restriction on delegation, the returned $\text{SK}_{\mathbf{Y}}$ is independent of the path taken.

Challenge. The adversary submits two messages m_0 and m_1 of equal length and a vector $\mathbf{x}^*$ with the restriction that it holds $\mathbf{x}^* \notin \mathsf{span}(\mathbf{Y})$ for any query $\mathbf{Y}$ in Phase 1. 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 as in Phase 1 with the restriction that it must hold $\mathbf{x}^* \notin \mathsf{span}(\mathbf{Y})$ for any query $\mathbf{Y}$.

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

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

Definition 2. *A spatial encryption scheme is* adaptively secure *if all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\text{SE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.*

8.3 Construction

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

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

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

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

$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m)$: On input a vector $\mathbf{x} := (x_1, \ldots, x_n)^\top \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^n$ and $m \in \mathbb{G}_T$, sample

$$(g_0, g_1, \ldots, g_n, g_{n+1}) \leftarrow \mathsf{SampG}(\mathrm{PP}; s),\ g'_T \leftarrow \mathsf{SampGT}(\mu(\mathrm{MSK}); s)$$

and output

$$\mathrm{CT}_{\mathbf{x}} := \left(C_0 := g_0,\ C_1 := g_{n+1} \cdot \prod_{i=1}^{n} g_i^{x_i},\ C_2 := g'_T \cdot m \right) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T.$$

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{Y})$: On input a space $\mathbf{Y} := (y_{i,j}) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^{n \times \ell}$, sample

$$(h_0, h_1, \ldots, h_n, h_{n+1}) \leftarrow \mathsf{SampH}(\mathrm{PP})$$

and output

$$\mathrm{SK}_{\mathbf{Y}} := \left(K_0 := h_0,\ K_1 := \mathrm{MSK} \cdot h_{n+1},\ K_2 := \prod_{i=1}^{n} h_i^{y_{i,1}},\ \ldots,\ K_{\ell+1} := \prod_{i=1}^{n} h_i^{y_{i,\ell}} \right) \in (\mathbb{H})^{\ell+2}.$$

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{Y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{x} \in \mathsf{span}(\mathbf{Y})$, run

$$\mathrm{SK}_{\mathbf{x}} := (K_0, K_1, K_2) \leftarrow \mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{Y}}, \mathbf{x}).$$

Compute

$$e(g_0, \mathrm{MSK}) \leftarrow e(C_0, K_1 \cdot K_2)/e(C_1, K_0),$$

and recover the message as

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

$\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{Y}}, \mathbf{Y}')$: On input a secret key $\mathrm{SK}_{\mathbf{Y}} := (K_0, K_1, K_2, \ldots, K_{\ell+1})$ and a space $\mathbf{Y}' := (y'_{i,j}) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^{n \times \ell'}$ where $\mathsf{span}(\mathbf{Y}') \subseteq \mathsf{span}(\mathbf{Y})$, compute $\mathbf{T} := (t_{j,k}) \in \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}^{\ell \times \ell'}$ such that $\mathbf{Y}' = \mathbf{Y}\mathbf{T}$ and

$$\widetilde{\mathrm{SK}}_{\mathbf{Y}'} := \left(K_0,\ K_1,\ \prod_{j=1}^{\ell} K_{j+1}^{t_{j,1}},\ \ldots,\ \prod_{j=1}^{\ell} K_{j+1}^{t_{j,\ell'}} \right).$$

and sample $\mathrm{SK}' \leftarrow \mathsf{KeyGen}(\mathrm{MPK}, 1, \mathbf{Y}')$. Output

$$\mathrm{SK}_{\mathbf{y}'} := \widetilde{\mathrm{SK}}_{\mathbf{y}'} \cdot \mathrm{SK}'$$

where $\cdot$ denotes entry-wise multiplication. Note that to generate or delegate a secret key for a vector $\mathbf{x}$, we delegate to $\mathsf{span}(\mathbf{x})$.

Delegation. Fix $\mathbf{Y}$ and $\mathbf{Y}'$ such that $\mathbf{Y}' \subseteq \mathbf{Y}$. Let $\widetilde{\mathrm{SK}}_{\mathbf{Y}'}$ and SK' be the values computed by $\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{Y}}, \mathbf{Y}')$. It is straight-forward to verify that $\widetilde{\mathrm{SK}}_{\mathbf{Y}'}$ lies in the support of $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{Y}')$:

$$\begin{aligned}
\widetilde{\mathrm{SK}}_{\mathbf{Y}'} &= \left(K_0,\ K_1,\ \prod_{j=1}^{\ell} K_{j+1}^{t_{j,1}},\ \ldots,\ \prod_{j=1}^{\ell} K_{j+1}^{t_{j,\ell'}} \right) \\
&= \left(h_0,\ \mathrm{MSK} \cdot h_{n+1},\ \prod_{j=1}^{\ell} (\prod_{i=1}^{n} h_i^{y_{i,j}})^{t_{j,1}},\ \ldots,\ \prod_{j=1}^{\ell} (\prod_{i=1}^{n} h_i^{y_{i,j}})^{t_{j,\ell'}} \right) \\
&= \left(h_0,\ \mathrm{MSK} \cdot h_{n+1},\ \prod_{i=1}^{n} \prod_{j=1}^{\ell} h_i^{y_{i,j} t_{j,1}},\ \ldots,\ \prod_{i=1}^{n} \prod_{j=1}^{\ell} h_i^{y_{i,j} t_{j,\ell'}} \right) \\
&= \left(h_0,\ \mathrm{MSK} \cdot h_{n+1},\ \prod_{i=1}^{n} h_i^{y'_{i,1}},\ \ldots,\ \prod_{i=1}^{n} h_i^{y'_{i,\ell'}} \right)
\end{aligned}$$

By linearity of KeyGen and the $\mathbb{H}$-subgroup property, multiplying by SK' re-randomizes the key and yields independence of the path taken.

Correctness. It suffices to establish correctness for $\mathbf{x} = \mathbf{y}$, where $\mathbf{y} \in \mathsf{span}(\mathbf{Y})$, using the delegation property. Observe that for $\mathrm{CT}_{\mathbf{x}}, \mathrm{SK}_{\mathbf{x}}$,

$$\begin{aligned}
e(C_0, K_1 \cdot K_2)/e(C_1, K_0) &= e\Big(g_0, \mathrm{MSK} \cdot (h_{n+1} \cdot \prod_{i=1}^{n} h_i^{x_i})\Big) \cdot e\Big(g_{n+1} \cdot \prod_{i=1}^{n} g_i^{x_i}, h_0\Big)^{-1} \\
&= e(g_0, \mathrm{MSK}) \cdot \Big(e(g_0, h_{n+1}) \cdot \prod_{i=1}^{n} e(g_0, h_i)^{x_i}\Big) \cdot \Big(e(g_{n+1}, h_0) \cdot \prod_{i=1}^{n} e(g_i, h_0)^{x_i}\Big)^{-1} \\
&= e(g_0, \mathrm{MSK})
\end{aligned}$$

where the last equality relies on *associative*, namely $e(g_0, h_i) = e(g_i, h_0)$ and $e(g_{n+1}, h_0) = e(g_0, h_{n+1})$. Finally, by *projective*, $g'_T = e(g_0, \mathrm{MSK})$. Correctness follows readily.

8.4 Proof of Security

We prove the following theorem:

Theorem 2. *Under the left and right subgroup indistinguishability (described in Section 3) and the additional requirement that* $\mathrm{ord}(\mathbb{H})$ *is prime, our spatial encryption scheme in Appendix 8.3 is adaptively secure (in the sense of Definition 2). More precisely, for any adversary* $\mathcal{A}$ *that makes at most* q *key queries against the spatial encryption scheme, there exist adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *such that:*

$$\mathsf{Adv}_{\mathcal{A}}(\lambda)^{\mathrm{SE}}(\lambda) \leq \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda) + q \cdot \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_2}(\lambda) + q \cdot \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_3}(\lambda),$$

and

$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n),$$

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

The proof follows via a series of games. To describe the games, we must first define semi-functional keys and ciphertexts.

Auxiliary algorithms. It is helpful to define the following algorithms:

$\widehat{\mathsf{Enc}}(\text{PP}, \mathbf{x}, m; \text{MSK}, \mathbf{t})$: On input $\mathbf{x} \in \mathbb{Z}_{\text{ord}(\mathbb{H})}^n$, $m \in \mathbb{G}_T$, and $\mathbf{t} := (T_0, T_1, \ldots, T_n, T_{n+1}) \in \mathbb{G}^{n+1}$, output

$$\text{CT}_{\mathbf{x}} := \left(C_0 := T_0,\ C_1 := T_{n+1} \cdot \prod_{i=1}^{n} T_i^{x_i},\ C_2 := e(T_0, \text{MSK}) \cdot m \right).$$

$\widehat{\mathsf{KeyGen}}(\text{PP}, h, \mathbf{Y}; \mathbf{t})$: On input $h \in \mathbb{H}$, $\mathbf{Y} := (y_{i,j}) \in \mathbb{Z}_{\text{ord}(\mathbb{H})}^{n \times \ell}$, and $\mathbf{t} := (T_0, T_1, \ldots, T_n, T_{n+1}) \in \mathbb{H}^{n+1}$, output

$$\text{SK}_{\mathbf{y}} := \left(K_0 := T_0,\ K_1 := h \cdot T_{n+1},\ K_2 := \prod_{i=1}^{n} T_i^{y_{i,1}},\ \ldots,\ K_{\ell+1} := \prod_{i=1}^{n} T_i^{y_{i,\ell}} \right).$$

Auxiliary distributions. Auxiliary distributions and game sequence is defined exactly as the proof of our HIBE scheme in Section 4.2, using the auxiliary algorithms as defined above.

8.5 Pseudo-Normal to Pseudo-SF Keys

We need the following statistical lemma for spatial encryption.

Lemma 13 (implicit in [11, 3]). *For any prime p, for all $\mathbf{x} \in \mathbb{Z}_p^n$ and $\mathbf{Y} \in \mathbb{Z}_p^{n \times \ell}$, where $\mathbf{x} \notin \mathsf{span}(\mathbf{Y})$, the following distribution is identically distributed to the uniform distribution over $\mathbb{Z}_p^{\ell+1}$:*

$$\{\mathbf{u}^\top \mathbf{x}, \mathbf{u}^\top \mathbf{Y}\}$$

where $\mathbf{u} \leftarrow_{\text{R}} \mathbb{Z}_p^n$.

Lemma 14 ($\mathsf{Game}_{2,i,1}$ to $\mathsf{Game}_{2,i,2}$). *For $i = 1, \ldots, q$:*

$$|\mathsf{Adv}_{2,i,1}(\lambda) - \mathsf{Adv}_{2,i,2}(\lambda)| = 0.$$

Proof. The proof starts out the same as that in Lemma 4, until the point where we expand the expressions for $\widehat{\mathsf{Enc}}$ and $\widehat{\mathsf{KeyGen}}$. Here, we obtain:

$$\widehat{\mathsf{Enc}}(\text{PP}, \mathbf{x}, 1; \text{MSK}, \hat{\mathbf{g}} \cdot \hat{\mathbf{g}}') = (\hat{g}_0, \hat{g}_{n+1} \cdot \prod_{i=1}^{n} \hat{g}_i^{x_i} \cdot (\hat{g}_0^{\gamma_{n+1} + \sum_{i=1}^{n} x_i \gamma_i}), e(\hat{g}_0, \text{MSK}))$$

$$\widehat{\mathsf{KeyGen}}(\text{PP}, 1, \mathbf{x}; \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}') = (\hat{h}_0, \hat{h}_{n+1} \cdot \hat{h}_0^{\gamma_{n+1}}, \prod_{i=1}^{n} \hat{g}_i^{y_{i,1}} \cdot (\hat{h}_0^{\sum_{i=1}^{n} y_{i,1} \gamma_i}), \ldots, \prod_{i=1}^{n} \hat{g}_i^{y_{i,\ell}} \cdot (\hat{h}_0^{\sum_{i=1}^{n} y_{i,\ell} \gamma_i}))$$

$$\widehat{\mathsf{KeyGen}}(\text{PP}, (h^*)^\alpha, \mathbf{Y}; \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}') = (\hat{h}_0, \hat{h}_{n+1} \cdot (h^*)^\alpha \cdot \hat{h}_0^{\gamma_{n+1}}, \prod_{i=1}^{n} \hat{g}_i^{y_{i,1}} \cdot (\hat{h}_0^{\sum_{i=1}^{n} y_{i,1} \gamma_i}), \ldots, \prod_{i=1}^{n} \hat{g}_i^{y_{i,\ell}} \cdot (\hat{h}_0^{\sum_{i=1}^{n} y_{i,\ell} \gamma_i}))$$

Since h^* lies in the group generated by $\hat{h}_0$, we may replace $(h^*)^\alpha$ by $(\hat{h}_0)^{\alpha'}$ and "for all α" by "for all α'" and obtain a stronger claim. Now, by focusing on the exponents of the terms involving $\hat{g}_0$ and $\hat{h}_0$, it suffices to show that for all α':

$$\{\gamma_{n+1} + \sum_{i=1}^{n} x_i \gamma_i, \gamma_{n+1}, \sum_{i=1}^{n} y_{i,1} \gamma_i, \ldots, \sum_{i=1}^{n} y_{i,\ell} \gamma_i\} \quad \text{and}$$

$$\{\gamma_{n+1} + \sum_{i=1}^{n} x_i \gamma_i, \alpha' + \gamma_{n+1}, \sum_{i=1}^{n} y_{i,1} \gamma_i, \ldots, \sum_{i=1}^{n} y_{i,\ell} \gamma_i\}$$

are identically distributed. The last statement follows readily from Lemma 13. □

Acknowledgments. We thank Allison Lewko for insightful discussions.

References

[1] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.

[2] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[3] C. Chen, Z. Zhang, and D. Feng. Fully secure doubly-spatial encryption under simple assumptions. In *ProvSec*, pages 253–263, 2012.

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

[5] J. Chen and H. Wee. Fully, (almost) tightly secure IBE from standard assumptions. IACR Cryptology ePrint Archive, Report 2013/803, 2013. Preliminary version in [4].

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

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

[8] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.

[9] C. S. Jutla and A. Roy. Shorter quasi-adaptive nizk proofs for linear subspaces. In *ASIACRYPT (1)*, pages 1–20, 2013.

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

[11] 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.

[12] 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.

[13] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.

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

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

[16] T. Okamoto and K. Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In *CANS*, pages 138–159, 2011. Also, Cryptology ePrint Archive, Report 2011/648.

[17] S. C. Ramanna, S. Chatterjee, and P. Sarkar. Variants of waters' dual system primitives using asymmetric pairings. In *Public Key Cryptography*, pages 298–315, 2012. Also, Cryptology ePrint Archive, Report 2012/057.

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

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

[20] H. Wee. Dual system encryption via predicate encodings. In *TCC*, pages 616–637, 2014.