Shorter IBE and Signatures via Asymmetric Pairings

Jie Chen[1], Hoon Wei Lim[1], San Ling[1], Huaxiong Wang[1], and Hoeteck Wee[2,1] ⋆

[1] Division of Mathematical Sciences
School of Physical & Mathematical Sciences
Nanyang Technological University, Singapore
[2] George Washington University, USA
s080001@e.ntu.edu.sg
{hoonwei,lingsan,hxwang}@ntu.edu.sg
hoeteck@gwu.edu

Abstract. We present efficient Identity-Based Encryption (IBE) and signature schemes under the Symmetric External Diffie-Hellman (SXDH) assumption in bilinear groups; our IBE scheme also achieves anonymity. In both the IBE and the signature schemes, all parameters have constant numbers of group elements, and are shorter than those of previous constructions based on Decisional Linear (DLIN) assumption. Our constructions use both dual system encryption (Waters, Crypto '09) and dual pairing vector spaces (Okamoto and Takashima, Pairing '08, Asiacrypt '09). Specifically, we show how to adapt the recent DLIN-based instantiations of Lewko (Eurocrypt '12) to the SXDH assumption. To our knowledge, this is the first work to instantiate either dual system encryption or dual pairing vector spaces under the SXDH assumption. Furthermore, our work could be extended to many other Functional Encryption. Particularly, we show how to instantiate our framework to Inner Product Encryption (IPE) and Key-Policy Functional Encryption (KP-FE). All parameters of our constructions are shorter than those of DLIN-based constructions.

⋆ Research of the authors is supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03. Hoeteck Wee's work is also supported by NSF CAREER Award CNS-1237429.

1 Introduction

Identity-Based Encryption. The idea of using a user's identity as her public encryption key, and thus eliminating the need for a public key certificate, was conceived by Shamir [36]. Such a primitive is known as Identity-Based Encryption (IBE), which has been extensively studied particularly over the last decade. We now have constructions of IBE schemes from a large class of assumptions, namely pairings, quadratic residuosity and lattices, starting with the early constructions in the random oracle model [9, 18, 24], to more recent constructions in the standard model [16, 7, 8, 17, 1].

Short IBE. It is desirable that an IBE scheme be as efficient as possible, if it were to have any impact on practical applications. Ideally, we would like to have constant-size public parameters, secret keys, and ciphertexts. Moreover, the scheme should ideally achieve full security, namely to be resilient even against an adversary that adaptively selects an identity to attack based on previous secret keys. The first fully secure efficient IBE with constant-size public parameters and ciphertexts under standard assumptions was obtained by Waters [39] in 2009; this scheme relied on the Decisional Bilinear Diffie-Hellman (DBDH) and Decisional Linear (DLIN) assumptions. Since then, Lewko and Waters [29] and Lewko [28] gave additional fully secure efficient IBE schemes that achieve incomparable guarantees. Prior to these works, all known IBEs (in the standard model) were either selectively secure [16, 7, 17, 1], or require long parameters [8, 38, 17, 1], or were based on less standard assumptions that depended on the query complexity of the adversary [23]. From a practical stand-point, Waters' fully secure IBE [39] is still not very efficient as it has relatively large ciphertexts and secret keys, i.e., eleven and nine group elements,[1] respectively. Lewko's scheme [28] improved on both of these parameters at the cost of larger public parameters and master key.

Shorter IBE? In his work, Waters also suggested obtaining even more efficient IBE schemes by turning to asymmetric bilinear groups:

> Using the SXDH assumption we might hope to shave off three group elements from both ciphertexts and private keys.

In fact, improving the efficiency of a scheme using asymmetric pairings was first observed by Boneh, Boyen and Shacham [10]. At a fixed security level, group elements in the asymmetric setting are smaller and pairings can be computed more efficiently [21]. (Estimated bit sizes of group elements for bilinear group generators are given in next paragraph.) Informally, the SXDH assumption states that there are prime-order groups (G_1, G_2, G_T) that admits a bilinear map $e : G_1 \times G_2 \rightarrow G_T$ such that the Decisional Diffie-Hellman (DDH) assumption holds in both G_1 and G_2. The SXDH assumption was formally defined by Ballard et al. [3] in their construction of a searchable encryption scheme, and has since been used in a number of different contexts, including secret-handshake schemes [2], anonymous IBE [19], continual leakage-resilience [14], and most notably, Groth-Sahai proofs [26]. Evidence for the validity of this assumption were presented in the works of Verheul [37] and Galbraith and Rotger [22].

[1] Here, we do not separately consider group elements from target groups of pairings, although a ciphertext typically has a group element that is from an associated target group. In Table 2, we give more accurate sizes comparing existing and our scheme.

Symmetric vs Asymmetric Pairings. The ordinary elliptic curves that give the best performance while providing discrete log security comparable to three commonly proposed levels of AES security are given in Table 1.

Pairings	80-bit AES			128-bit AES			256-bit AES		
	G_1	G_2	G_T	G_1	G_2	G_T	G_1	G_2	G_T
Asymmetric	170	340	1020	256	512	3072	640	2560	15360
Symmetric	176	176	1056	512	512	3072	2560	2560	15360

Table 1. Estimated bit sizes of elements in bilinear groups. The group sizes follow the 2007 NIST recommendations [4], descriptions of the elliptic curves are in [20]: 80-bit security, a 170-bit MNT curve [31] with embedding degree $k = 6$; 128-bit security, a 256-bit Barreto-Naehrig curve [5] with $k = 12$; 256-bit security, a 640-bit Brezing-Weng curve [15] with $k = 24$.

Note that we assume that curves that support sextic twists are used for $k = 12$ and $k = 24$ as this allows elements of G_2 to be 1/6 the size of elements of G_T. We also assume that point compression is used to represent a group element. We further note that a symmetric pairing only exists on supersingular elliptic curves. The restriction to supersingular elliptic curves means that at high security levels the group G_1 will be much larger than the group G_1 on an equivalent ordinary curve.

1.1 Our Contributions

In this work, we present a more efficient IBE scheme under the SXDH assumption; our scheme also achieves anonymity.[2] The ciphertexts and secret keys consist of only five and four group elements, respectively. That is, we shave off two group elements from both ciphertexts and private keys in Lewko's DLIN-based IBE [28]. Table 2 gives a summary of comparisons between existing and our IBE schemes. Applying Naor's transform [9, 12] to our scheme, we also obtain an efficient signature scheme.

Source	$\|\mathsf{PP}\|$	$\|\mathsf{SK}\|$	$\|\mathsf{CT}\|$	# pairing	anonymity	assumptions
Waters [38]	$(4+\lambda)\|G_0\|$	$2\|G_0\|$	$2\|G_0\| + \|G_T\|$	2	No	DBDH
Waters [39]	$12\|G_0\| + \|G_T\|$	$8\|G_0\| + \|\mathbb{Z}_q\|$	$9\|G_0\| + \|G_T\| + \|\mathbb{Z}_q\|$	9	No	DLIN DBDH
Lewko [28]	$24\|G_1\| + \|G_T\|$	$6\|G_2\|$	$6\|G_1\| + \|G_T\|$	6	Yes	DLIN
RCS [35]	$8\|G_1\| + \|G_T\|$	$6\|G_2\| + \|\mathbb{Z}_q\|$	$8\|G_1\| + \|G_T\|$	7	No	XDH DLIN DBDH
Ours	$8\|G_1\| + \|G_T\|$	$4\|G_2\|$	$4\|G_1\| + \|G_T\|$	4	Yes	SXDH

Table 2. Comparison between existing and our IBE schemes, where λ is the security parameter (and it depends on the curve we use). Here, $|\mathsf{PP}|$, $|\mathsf{SK}|$, $|\mathsf{CT}|$, # pairing stand for public parameters size, secret key size, ciphertext size, the number of pairing for decryption, respectively; $|G_{\mathbf{x}}|$ represents bit length of group $G_{\mathbf{x}}$, where $\mathbf{x} \in \{0, 1, 2, T\}$, and G_0 refers to a group in the symmetric pairing setting.

[2] It follows from our analysis that Lewko's IBE [28] is also anonymous, although this was not pointed out in her paper.

Our approach. As with all known fully secure efficient IBEs, our construction relies on Waters' dual system encryption framework [39]. Following Lewko's DLIN-based IBE [28], we instantiate dual system encryption under the SXDH assumption via dual pairing vector spaces [32, 33], which is a technique to achieve orthogonality in prime-order groups. This is the first work to instantiate either dual system encryption or dual pairing vector spaces under the SXDH assumption. We proceed to highlight several salient features of our IBE scheme in relation to Lewko's IBE [28]:

- Our scheme has an extremely simple structure, similar to the selectively secure IBE of Boneh and Boyen [7], as well as the fully secure analogues given by Lewko and Waters [29] and Lewko [28].
- By shifting from the DLIN assumption to the simpler SXDH assumption, we obtain an IBE scheme that is syntactically simpler and achieves shorter parameters. Specifically, Lewko's IBE scheme [28] relies on 6 basis vectors to simulate the subgroup structure in the Lewko-Waters IBE scheme [29], whereas our construction uses only 4 basis vectors. This means that we can use a 4-dimensional vector space instead of a 6-dimensional one. As a result, we save two group elements in both the secret key and the ciphertext, that is, by a factor of 1/3. The savings for the public parameters and master key is even more substantial, because we use only two basis vectors for the main scheme, as opposed to four basis vectors in Lewko's scheme. In both our scheme and in Lewko's, the remaining two basis vectors are used for the semi-functional components in the proof of security.
- The final step of the proof of security (after switching to semi-functional secret keys and ciphertexts) is different from that of Lewko's. We rely on an information theoretic argument similar to that in [34] instead of computational arguments.

Finally, we believe that our SXDH instantiation constitutes a simpler demonstration of the power of dual pairing vector spaces. We also show how to instantiate our framework to Inner Product Encryption (IPE) [27] and Key-Policy Functional Encryption (KP-FE) [34]. All parameters of our constructions are shorter than those of DLIN-based constructions [34]. Table 3 gives a summary of comparisons between the IPE/KP-FE schemes of [34] and ours.

Source		$\|\mathsf{PP}\|$	$\|\mathsf{SK}\|$	$\|\mathsf{CT}\|$	# pairing	assumptions
IPE	OT [34]	$3n^2\|G_0\| + 1\|G_T\|$	$3n\|G_0\|$	$3n\|G_0\| + 1\|G_T\|$	$3n$	DLIN
	Ours	$2n^2\|G_1\| + 1\|G_T\|$	$2n\|G_2\|$	$2n\|G_1\| + 1\|G_T\|$	$2n$	SXDH
KP-FE	OT [34]	$3n^2d\|G_0\| + 1\|G_T\|$	$3n\hat{a}\|G_0\|$	$3nd\|G_0\| + 1\|G_T\|$	$3n\hat{a}$	DLIN
	Ours	$2n^2d\|G_1\| + 1\|G_T\|$	$2n\hat{a}\|G_2\|$	$2nd\|G_1\| + 1\|G_T\|$	$2n\hat{a}$	SXDH

Table 3. Comparison between the IPE/KP-FE schemes of [34] and ours. All measurements are rough estimations (after removing small terms). Here, n refers to the dimension parameter in IPE setting or the parameter for the maximal dimension of attribute vector in KP-FE setting; d denotes size of the attribute set; and $\hat{a}$ is the number of rows in the matrix of the access structure.

Independent work of Ramanna et al. An independent work of Ramanna, Chatterjee and Sarkar [35] also demonstrated how to obtain more efficient fully secure IBE via asymmetric pairings. Similar to our work, their constructions rely on dual system encryption; however, they do not make use of dual pairing vector spaces. Our constructions achieve shorter ciphertexts and secret keys than

their work, while relying on a single assumption (whereas their construction relies on a triplet of assumptions). Moreover, our scheme achieves anonymity; theirs does not. Finally, they obtain their schemes via careful optimizations, whereas our scheme is derived via a more general framework.

2 Preliminaries

In what follows, we borrow the definition and the game-based security model for Functional Encryption (FE) from [13] which are adequate to define all encryption systems in this paper.

2.1 Functional Encryption

As in [13], we first describe a functionality $\hat{F}$ of the syntactic definition of FE. The functionality $\hat{F}$ describes the functions of a plaintext that can be learned from the ciphertext:

Definition 1. *A functionality $\hat{F}$ defined over $(\mathcal{K}, \mathcal{X})$ is a function $\hat{F} : \mathcal{K} \times \mathcal{X} \to \{0,1\}^*$ described as a (deterministic) Turing Machine. The set $\mathcal{K}$ is called the key space and the set $\mathcal{X}$ is called the plaintext space. We require that the key space $\mathcal{K}$ contain a special key called the empty key denoted ϱ.*

An FE scheme for the functionality $\hat{F}$ enables one to evaluate $\hat{F}(\mathrm{v}, \mathrm{x})$ given the encryption of x and a secret key SK_v for v. The algorithm for evaluation $\hat{F}(\mathrm{v}, \mathrm{x})$ using SK_v is called *decrypt*. More precisely, an FE scheme is defined as follows:

Definition 2. *A functional encryption scheme (FE) for a functionality $\hat{F}$ defined over $(\mathcal{K}, \mathcal{X})$ is a tuple of four probabilistic polynomial-time (PPT) algorithms (*Setup, KeyGen, Enc, Dec*) satisfying the following correctness condition for all $\mathrm{v} \in \mathcal{K}$ and $\mathrm{x} \in \mathcal{X}$:*

$(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$	(*generate a public and master secret key pair*)
$\mathsf{SK}_\mathrm{v} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \mathrm{v})$	(*generate a secret key for* v)
$\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathrm{x})$	(*encrypt plaintext* x)
$\mathrm{y} \leftarrow \mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_\mathrm{v}, \mathsf{CT})$	(*use SK_v to compute $\hat{F}(\mathrm{v}, \mathrm{x})$ from* CT)

then we require that $\mathrm{y} = \hat{F}(\mathrm{v}, \mathrm{x})$ with probability 1.

The empty key ϱ: The special key ϱ in $\mathcal{K}$ captures all the information about the plaintext that intentionally leaks from the ciphertext. The secret key for ϱ is empty and also denoted by ϱ. Thus, anyone can run $\mathsf{Dec}(\mathsf{PP}, \varrho, \mathsf{CT})$ on a ciphertext $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathrm{x})$ and obtain all the information about x that intentionally leaks from CT. Take IBE for example, $\hat{F}(\varrho, (\mathrm{id}, \mathrm{m}))$ outputs only $|\mathrm{m}|$ (the length of message m) in the attribute-hiding setting while it outputs $|\mathrm{m}|$ and the identity id in the payload-hiding setting. Henceforth, we assume that every FE scheme contains the empty key ϱ in the key space $\mathcal{K}$ and we will not explicitly mention it.

We now define the security model for FE. For the plaintext pair $(\mathrm{x}_0, \mathrm{x}_1)$ of an adversary's choice, we need the following requirement to make the experiment non-trivial:

$$\hat{F}(\mathrm{v}, \mathrm{x}_0) = \hat{F}(\mathrm{v}, \mathrm{x}_1) \text{ for all v for which the adversary has } \mathsf{SK}_\mathrm{v}. \tag{1}$$

Then we define a security game for an FE scheme as follows:

Definition 3. *For $\beta = 0, 1$ define an experiment β for an adversary $\mathcal{A}$ as follows:*

- Setup*: It runs* $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and gives* PP *to* $\mathcal{A}$.
- Query*:* $\mathcal{A}$ *adaptively submits key queries* v_i *in* $\mathcal{K}$ *for* $i = 1, 2, \ldots$ *and is given* $\mathsf{SK}_{\mathrm{v}_i} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \mathrm{v}_i)$.
- $\mathsf{Challenge}$*:* $\mathcal{A}$ *submits two plaintexts* $\mathrm{x}_0, \mathrm{x}_1 \in \mathcal{X}$ *satisfying requirement (1) and in return, it receives* $\mathsf{Enc}(\mathsf{PP}, \mathrm{x}_\beta)$.
- Guess*:* $\mathcal{A}$ *continues to issue key queries as before subject to requirement (1) and eventually outputs a bit in* $\{0, 1\}$.

For $\beta = 0, 1$ let W_β be the event that the adversary outputs 1 in Experiment β and define

$$\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) := |\Pr[W_0] - \Pr[W_1]|.$$

Definition 4. *An FE scheme is fully secure if for all PPT adversaries $\mathcal{A}$ the function $\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda)$ is negligible.*

In all encryption systems of this paper, a plaintext $\mathrm{x} \in \mathcal{X}$ is itself a pair $(\mathrm{ind}, \mathrm{m}) \in \mathcal{I} \times \mathcal{M}$ where ind is called an index and m is called the payload message. Let $\mathrm{x}_0 = (\mathrm{ind}_0, \mathrm{m}_0)$, $\mathrm{x}_1 = (\mathrm{ind}_1, \mathrm{m}_1) \in \mathcal{X}$ be the adversary's choice of plaintext pair, we then consider the following variations:

- If the adversary's choice subjects to the restriction that $\mathrm{ind}_0 = \mathrm{ind}_1$, the security game is then under the *payload-hiding* model;
- If the adversary's queries subject to the restriction that $\hat{F}(\mathrm{v}_i, (\mathrm{ind}_0, \mathrm{m}_0)) \neq \mathrm{m}_0$ and $\hat{F}(\mathrm{v}_i, (\mathrm{ind}_1, \mathrm{m}_1)) \neq \mathrm{m}_1$ for all the key queries v_i, the security game is then under the *weakly attribute-hiding* (or *anonymous*) model.

2.2 Identity-Based Encryption

In the IBE setting, a functionality $\hat{F}$ is defined over a key space and an index space using sets of identities. The key space $\mathcal{K}$ and index space $\mathcal{I}$ for IBE then corresponds to all identities id. Here

$$\hat{F}(\mathrm{id}, (\mathrm{id}', \mathrm{m})) := \begin{cases} \mathrm{m} & \text{if } \mathrm{id}' = \mathrm{id} \\ \perp & \text{otherwise.} \end{cases}$$

2.3 Inner Product Encryption

In the IPE setting, a functionality $\hat{F}$ is defined over a key space and an index space using sets of vectors. The key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) for IPE then corresponds to all non-zero vectors $\mathbf{v}$ (resp. $\mathbf{x}$)). Here

$$\hat{F}(\mathbf{v}, (\mathbf{x}, \mathrm{m})) := \begin{cases} \mathrm{m} & \text{if } \mathbf{x} \cdot \mathbf{v} = 0 \\ \perp & \text{otherwise.} \end{cases}$$

2.4 Key-Policy Functional Encryption

We first describe the concept of span programs typically required by ABE.

Definition 5 (Span Programs [6]). *Let $\{p_1, \ldots, p_n\}$ be a set of variables. A span program over $\mathbb{Z}_q$ is a labeled matrix $(\hat{\mathbf{A}}, \hat{\rho})$ where $\hat{\mathbf{A}}$ is an $(\hat{a} \times \hat{b})$ matrix over $\mathbb{Z}_q$ and $\hat{\rho}$ is a labeling of the rows of $\hat{\mathbf{A}}$ by literals from $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$ (every row is labeled by one literal), i.e., $\hat{\rho} : [\hat{a}] \to \{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$.*

A span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0,1\}^n$ define the submatrix $\hat{\mathbf{A}}_\delta$ of $\hat{\mathbf{A}}$ consisting of those rows whose labels are set to 1 by the input, i.e., either rows labeled by some p_i such that $\delta_i = 1$ or rows labeled by some $\neg p_i$ such that $\delta_i = 0$. (i.e., $\hat{\gamma} : [\hat{a}] \to \{0,1\}$ is defined by $\hat{\gamma}(j) = 1$ if $[\hat{\rho}(j) = p_i] \wedge [\delta_i = 1]$ or $[\hat{\rho}(j) = \neg p_i] \wedge [\delta_i = 0]$, and $\hat{\gamma}(j) = 0$ otherwise. Let $\hat{\mathbf{A}}_\delta := (\hat{\mathbf{A}}_j)_{\hat{\gamma}(j)=1}$, where $\hat{\mathbf{A}}_j$ is the j-th row of $\hat{\mathbf{A}}$.)

The span program $(\hat{\mathbf{A}}, \hat{\rho})$ accepts δ if and only if $\mathbf{1} \in \mathsf{span}\langle \hat{\mathbf{A}}_\delta \rangle$, i.e., some linear combination of the rows of $\hat{\mathbf{A}}_\delta$ gives the all one vector $\mathbf{1}$, where $\mathbf{1} = (1, \ldots, 1)$. A span program computes a Boolean function $\hat{f}$ if it accepts exactly those inputs δ where $\hat{f}(\delta) = 1$.

A span program is called monotone if the labels of the rows are only the positive literals $\{p_1, \ldots, p_n\}$. Otherwise, it is non-monotone.

We first give the notion of a non-monotone access structure with evaluating map γ by using inner-products of attribute vectors.

Definition 6 (Inner Products of Attribute Vectors and Access Structures [34]). *$\mathcal{U}_i$ ($i = 1, \ldots, d$ and $\mathcal{U}_i \subset \{0,1\}^*$) is a sub-universe, a set of attributes, each of which is expressed by a pair of sub-universe id and n_i-dimensional vector, i.e., $(i, \mathbf{v})$, where $i \in [d]$ and $\mathbf{v} \in \mathbb{Z}_q^{n_i} \backslash \{\mathbf{0}\}$. We denote such structure as $\boldsymbol{n} := (d; n_1, \ldots, n_d)$.*

We define such an attribute to be a variable p of a span program $(\hat{\mathbf{A}}, \hat{\rho})$, i.e., $p := (i, \mathbf{x})$. An access structure $\mathbb{A}$ is a span program $(\hat{\mathbf{A}}, \hat{\rho})$ along with variables $p := (i, \mathbf{x})$, $p' := (i', \mathbf{x}')$, ..., i.e., $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ such that $\hat{\rho} : [\hat{a}] \to \{(i, \mathbf{x}), (i', \mathbf{x}'), \ldots, \neg(i, \mathbf{x}), \neg(i', \mathbf{x}'), \ldots\}$.

Let Γ be a set of attributes, i.e., $\Gamma := \{(i, \mathbf{v}_i) | \mathbf{v}_i \in \mathbb{Z}_q^{n_i} \backslash \{\mathbf{0}\}, 1 \leq i \leq d\}$, where $1 \leq i \leq d$ means that i is an element of some subset of $[d]$.

When Γ is given the access structure $\mathbb{A}$, map $\hat{\gamma} : [\hat{a}] \to \{0,1\}$ for span program $(\hat{\mathbf{A}}, \hat{\rho})$ is defined as follows: For all $j \in [\hat{a}]$, set $\hat{\gamma}(j) = 1$ if $[\hat{\rho}(j) = (i, \mathbf{x}_j)] \wedge [(i, \mathbf{v}_i) \in \Gamma] \wedge [\mathbf{x}_j \cdot \mathbf{v}_i = 0]$ or $[\hat{\rho}(j) = \neg(i, \mathbf{x}_j)] \wedge [(i, \mathbf{v}_i) \in \Gamma] \wedge [\mathbf{x}_j \cdot \mathbf{v}_i \neq 0]$. Set $\hat{\gamma}(j) = 0$ otherwise.

Access structure $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ accepts Γ iff $\mathbf{1} \in \mathsf{span}\langle (\hat{\mathbf{A}}_j)_{\hat{\gamma}(j)=1} \rangle$.

We use the following secret-sharing scheme for a non-monotone access structure or span program.

Definition 7. *A secret-sharing scheme for access structure $\mathbb{A}$ is a linear secret-sharing scheme (LSSS) in $\mathbb{Z}_q$ and is represented by $(\hat{\mathbf{A}}, \hat{\rho})$ if it consists of two efficient algorithms:*

Lin.Share$_{(\hat{\mathbf{A}}, \hat{\rho})}$ *: Let $\hat{\mathbf{A}}$ be $\hat{a} \times \hat{b}$ share-generating matrix. Let $\mathbf{f} := (w_1, \ldots, w_{\hat{b}}) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\hat{b}}$. Then, $s_0 := \mathbf{1} \cdot \mathbf{w}^\top$ is the secret to be shared, and $\mathbf{s}^\top := (s_1, \ldots, s_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \mathbf{w}^\top$ is the vector of $\hat{a}$ shares of the secret s_0 and the share s_j belongs to $\hat{\rho}(j)$.*

Lin.Recon$_{(\hat{\mathbf{A}}, \hat{\rho})}$ *: If the span program $(\hat{\mathbf{A}}, \hat{\rho})$ accept δ, or access structure $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ accepts Γ, i.e., $\mathbf{1} \in \mathsf{span}\langle (\hat{\mathbf{A}}_j)_{\hat{\gamma}(j)=1} \rangle$ with $\hat{\gamma} : [\hat{a}] \to \{0,1\}$, then there exist constants $\{\alpha_j \in \mathbb{Z}_q | j \in \Pi\}$ such that $\Pi \subseteq \{j \in [\hat{a}] | \hat{\gamma}(j) = 1\}$ and $\Sigma_{j \in \Pi} \alpha_j s_j = s_0$. Furthermore, these constants $\{\alpha_j\}$ can be computed in time polynomial in the size of matrix $\hat{\mathbf{A}}$.*

In a KP-FE scheme supporting non-monotone access structure, a functionality $\hat{F}$ is defined over a key space and an index space using sets of non-monotone access structures and attribute vector tuples, respectively (see Definition 6). The key space $\mathcal{K}$ corresponds to all non-monotone access structures $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$, while the index space $\mathcal{I}$ corresponds to all attribute sets Γ. Here,

$$\hat{F}(\mathbb{A}, (\Gamma, \mathrm{m})) := \begin{cases} \mathrm{m} & \text{if } \mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho}) \text{ accepts } \Gamma \\ \perp & \text{otherwise.} \end{cases}$$

2.5 Signatures

A signature scheme is made up of three algorithms, $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ for generating keys, signing, and verifying signatures, respectively.

- $\mathsf{KeyGen}(1^\lambda)$ The key generation algorithm takes in the security parameter λ, and outputs the public key PK, and the secret key SK.
- $\mathsf{Sign}(\mathsf{SK}, \mathrm{m})$ The signing algorithm takes in the secret key SK, and a message M, and produces a signature σ for that message.
- $\mathsf{Verify}(\mathsf{PK}, \sigma, \mathrm{m})$ The verifying algorithm takes in the public key PK, and a signature pair (σ, m), and outputs `valid` or `invalid`.

The standard notion of security for a signature scheme is called existential unforgeability under a chosen message attack [25], which is defined using the following game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$.

- Setup The challenger $\mathcal{B}$ runs the setup algorithm to generate PK and SK. It gives PK to the adversary $\mathcal{A}$.
- Query The adversary $\mathcal{A}$ adaptively requests for messages $\mathrm{m}_1, \ldots, \mathrm{m}_{q_n} \in \{0,1\}^*$, and is provided with corresponding signatures $\sigma_1, \ldots, \sigma_{q_n}$ by running the sign algorithm Sign.
- Output Eventually, the adversary $\mathcal{A}$ outputs a pair (m, σ).

The advantage $\mathsf{Adv}^{\mathsf{Sig}}_{\mathcal{A}}(\lambda)$ of an adversary $\mathcal{A}$ is defined to be the probability that $\mathcal{A}$ wins in the above game, namely

(1) m is not any of $\mathrm{m}_1, \ldots, \mathrm{m}_{q_n}$;
(2) $\mathsf{Verify}(\mathsf{PK}, \sigma, \mathrm{m})$ outputs `valid`.

Definition 8. *A signature scheme is existentially unforgeable under an adaptive chosen message attack if all PPT adversaries achieve at most a negligible advantage in the above security game.*

We assume that for any PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ wins in the above game is negligible in the security parameter λ.

2.6 Dual Pairing Vector Spaces

Our constructions are based on dual pairing vector spaces proposed by Okamoto and Takashima [32, 33]. In this paper, we concentrate on the asymmetric version [34]. We only briefly describe how to generate random dual orthonormal bases. See [32, 33, 34] for a full definition of dual pairing vector spaces.

Definition 9 (Asymmetric bilinear pairing groups). *Asymmetric bilinear pairing groups* $(q, G_1, G_2, G_T, g_1, g_2, e)$ *are a tuple of a prime* q*, cyclic (multiplicative) groups* G_1, G_2 *and* G_T *of order* q*,* $g_1 \neq 1 \in G_1$*,* $g_2 \neq 1 \in G_2$*, and a polynomial-time computable nondegenerate bilinear pairing* $e : G_1 \times G_2 \to G_T$ *i.e.,* $e(g_1^s, g_2^t) = e(g_1, g_2)^{st}$ *and* $e(g_1, g_2) \neq 1$*.*

In addition to referring to individual elements of G_1 or G_2, we will also consider "vectors" of group elements. For $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_q^n$ and $g_\beta \in G_\beta$, we write $g_\beta^{\mathbf{v}}$ to denote a n-tuple of elements of G_β for $\beta = 1, 2$:

$$g_\beta^{\mathbf{v}} := (g_\beta^{v_1}, \ldots, g_\beta^{v_n}).$$

For any $a \in \mathbb{Z}_q$ and $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_q^n$, we have:

$$g_\beta^{a\mathbf{v}} := (g_\beta^{av_1}, \ldots, g_\beta^{av_n}), \quad g_\beta^{\mathbf{v}+\mathbf{w}} := (g_\beta^{v_1+w_1}, \ldots, g_\beta^{v_n+w_n}).$$

Then we define

$$e(g_1^{\mathbf{v}}, g_2^{\mathbf{w}}) := \prod_{i=1}^{n} e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\mathbf{v}\cdot\mathbf{w}}.$$

Here, the dot product is taken modulo q.

Dual Pairing Vector Spaces. For a fixed (constant) dimension n, we will choose two random bases $\mathbb{B} := (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ and $\mathbb{B}^* := (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ of $\mathbb{Z}_q^n$, subject to the constraint that they are "dual orthonormal", meaning that

$$\mathbf{b}_j \cdot \mathbf{b}_k^* = 0 \,(\mathsf{mod}\ q)$$

whenever $j \neq k$, and

$$\mathbf{b}_j \cdot \mathbf{b}_j^* = \psi \,(\mathsf{mod}\ q)$$

for all j, where ψ is a random element of $\mathbb{Z}_q$. We denote such algorithm as $\mathsf{Dual}(\mathbb{Z}_q^n)$.

Then for generators $g_1 \in G_1$ and $g_2 \in G_2$, we have

$$e(g_1^{\mathbf{b}_j}, g_2^{\mathbf{b}_k^*}) = 1$$

whenever $j \neq k$, where 1 here denotes the identity element in G_T.

More generally, we can sample multiple tuple of "dual orthonormal" bases. Namely, for fixed (constant) dimension $n_1, \ldots, n_d$, we will choose d tuples of two random bases $\mathbb{B}_i := (\mathbf{b}_{1,i}, \ldots, \mathbf{b}_{n_i,i})$ and $\mathbb{B}_i^* := (\mathbf{b}_{1,i}^*, \ldots, \mathbf{b}_{n_i,i}^*)$ of $\mathbb{Z}_q^{n_i}$, subject to the constraint that they are "dual orthonormal", meaning that

$$\mathbf{b}_{j,i} \cdot \mathbf{b}_{k,i}^* = 0 \,(\mathsf{mod}\ q)$$

whenever $j \neq k$, and

$$\mathbf{b}_{j,i} \cdot \mathbf{b}_{j,i}^* = \psi \,(\mathsf{mod}\ q)$$

for all j and i, where ψ is a random element of $\mathbb{Z}_q$. We denote such algorithm as $\mathsf{Dual}(\mathbb{Z}_q^{n_1}, \ldots, \mathbb{Z}_q^{n_d})$.

2.7 SXDH Assumptions

Definition 10 (DDH1: Decisional Diffie-Hellman Assumption in G_1). *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned}&\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathrm{R}} \mathcal{G},\\ &a, b, c \leftarrow_{\mathrm{R}} \mathbb{Z}_q,\\ &D := (\mathbb{G}; g_1, g_2, g_1^a, g_1^b).\end{aligned}$$

We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0,1\}$),

$$\mathsf{Adv}^{\mathsf{DDH1}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(D, g_1^{ab}) - \Pr[\mathcal{A}(D, g_1^{ab+c})]\right|.$$

is negligible in the security parameter λ.

The dual of above assumption is Decisional Diffie-Hellman assumption in G_2 (denoted as DDH2), which is identical to Definition 10 with the roles of G_1 and G_2 reversed. We say that:

Definition 11. *The Symmetric External Diffie-Hellman assumption holds if DDH problems are intractable in both G_1 and G_2.*

2.8 Statistical Indistinguishability Lemma

We require the following lemma for our security proofs, which is derived from [34].

Lemma 1. *For $p \in \mathbb{Z}_q$, let $C_p := \left\{(\mathbf{x}, \mathbf{v}) | \mathbf{x} \cdot \mathbf{v} = p, \mathbf{0} \neq \mathbf{x}, \mathbf{0} \neq \mathbf{v} \in \mathbb{Z}_q^n\right\}$. For all $(\mathbf{x}, \mathbf{v}) \in C_p$, for all $(\mathbf{z}, \mathbf{w}) \in C_p$, and $\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n \times n}$ ($\mathbf{A}$ is invertible with overwhelming probability),*

$$\Pr[\mathbf{x}\mathbf{A}^{\top} = \mathbf{z} \ \wedge \ \mathbf{v}\mathbf{A}^{-1} = \mathbf{w}] = \frac{1}{\#C_p}.$$

3 Subspace Assumptions via SXDH

In this section, we present Subspace assumptions derived from the SXDH assumption. We will rely on these assumptions later to instantiate our encryption schemes. These are analogues of the DLIN-based Subspace assumptions given in [28, 34].

3.1 Decisional Subspace Assumption

Definition 12 (DS1: Decisional Subspace Assumption in G_1). *Given a group generator $\mathcal{G}(\cdot)$, define the following distribution:*

$$\begin{aligned}&\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathrm{R}} \mathcal{G}(1^{\lambda}),\\ &(\mathbb{B}, \mathbb{B}^*) \leftarrow_{\mathrm{R}} \mathsf{Dual}(\mathbb{Z}_q^N); \tau_1, \tau_2, \mu_1, \mu_2 \leftarrow_{\mathrm{R}} \mathbb{Z}_q,\\ &U_1 := g_2^{\mu_1 \mathbf{b}_1^* + \mu_2 \mathbf{b}_{K+1}^*}, \ldots, U_K := g_2^{\mu_1 \mathbf{b}_K^* + \mu_2 \mathbf{b}_{2K}^*},\\ &V_1 := g_1^{\tau_1 \mathbf{b}_1}, \ldots, V_K := g_1^{\tau_1 \mathbf{b}_K},\\ &W_1 := g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{K+1}}, \ldots, W_K := g_1^{\tau_1 \mathbf{b}_K + \tau_2 \mathbf{b}_{2K}},\\ D := (\mathbb{G}; &g_2^{\mathbf{b}_1^*}, \ldots, g_2^{\mathbf{b}_K^*}, g_2^{\mathbf{b}_{2K+1}^*}, \ldots, g_2^{\mathbf{b}_N^*}, g_1^{\mathbf{b}_1}, \ldots, g_1^{\mathbf{b}_N}, U_1, \ldots, U_K, \mu_2)\end{aligned}$$

where K, N are fixed positive integers that satisfy $2K \leq N$. We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0,1\}$),

$$\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, V_1, \ldots, V_K) = 1] - \Pr[\mathcal{A}(D, W_1, \ldots, W_K) = 1]|$$

is negligible in the security parameter λ.

Lemma 2. *If the DDH assumption in G_1 holds, then the Subspace assumption in G_1 stated in Definition 12 also holds. More precisely, for any adversary $\mathcal{A}$ against the Subspace assumption in G_1, there exist probabilistic algorithms $\mathcal{B}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH1}}_{\mathcal{B}}(\lambda).$$

Proof. We assume there exists a PPT algorithm $\mathcal{A}$ breaking the Subspace assumption with non-negligible advantage $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{A}}(\lambda)$ (for some fixed positive integers K, N satisfying $N \geq 2K$). We create a PPT algorithm $\mathcal{B}$ which breaks the DDH assumption in G_1 with non-negligible advantage $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{A}}(\lambda)$. $\mathcal{B}$ is given $g_1, g_2, g_1^a, g_1^b, T$, where T is either g_1^{ab} or T is a uniformly random element of G_1.

$\mathcal{B}$ first samples random dual orthonormal bases, denoted by $\mathbf{f}_1, \ldots, \mathbf{f}_N$ and $\mathbf{f}^*_1, \ldots, \mathbf{f}^*_N$. From the definition, $\mathcal{B}$ chooses vectors $\mathbf{f}_1, \ldots, \mathbf{f}_N, \mathbf{f}^*_1, \ldots, \mathbf{f}^*_N$ randomly, subject to the constraints that $\mathbf{f}_i \cdot \mathbf{f}^*_j \equiv 0 \,(\mathsf{mod}\ q)$ when $j \neq k$, and $\mathbf{f}_j \cdot \mathbf{f}^*_j \equiv \psi \,(\mathsf{mod}\ q)$ for all j from 1 to N, where ψ is a random element of $\mathbb{Z}_q$. Then, $\mathcal{B}$ implicitly sets:

$$\begin{aligned}
&\mathbf{b}_1 := \mathbf{f}_1 + a\mathbf{f}_{K+1}, \ldots, \mathbf{b}_K := \mathbf{f}_K + a\mathbf{f}_{2K}, \\
&\mathbf{b}_{K+1} := \mathbf{f}_{K+1}, \ldots, \mathbf{b}_N := \mathbf{f}_N.
\end{aligned}$$

$\mathcal{B}$ also sets the dual basis as:

$$\begin{aligned}
&\mathbf{b}^*_1 := \mathbf{f}^*_1, \ldots, \mathbf{b}^*_K := \mathbf{f}^*_K, \\
&\mathbf{b}^*_{K+1} := \mathbf{f}^*_{K+1} - a\mathbf{f}^*_1, \ldots, \mathbf{b}^*_{2K} := \mathbf{f}^*_{2K} - a\mathbf{f}^*_K, \\
&\mathbf{b}^*_{2K+1} := \mathbf{f}^*_{2K+1}, \ldots, \mathbf{b}^*_N := \mathbf{f}^*_N.
\end{aligned}$$

We observe that under these definitions, $\mathbf{b}_j \cdot \mathbf{b}^*_k \equiv 0 \,(\mathsf{mod}\ q)$ when $j \neq k$, and $\mathbf{b}_j \cdot \mathbf{b}^*_j \equiv \psi \,(\mathsf{mod}\ q)$ for all j from 1 to N. We note that $\mathcal{B}$ can produce all of $g_1^{\mathbf{b}_1}, \ldots, g_1^{\mathbf{b}_N}$ (given g_1, g_1^a) as well as $g_2^{\mathbf{b}^*_1}, \ldots, g_2^{\mathbf{b}^*_K}$ and $g_2^{\mathbf{b}^*_{2K+1}}, \ldots, g_2^{\mathbf{b}^*_N}$ (given g_2). However, $\mathcal{B}$ cannot produce $g_2^{\mathbf{b}^*_{K+1}}, \ldots, g_2^{\mathbf{b}^*_{2K}}$ (these require knowledge of g_2^a). It is not difficult to check that $\mathbf{b}_1, \ldots, \mathbf{b}_N$ and $\mathbf{b}^*_1, \ldots, \mathbf{b}^*_N$ are properly distributed.

Now $\mathcal{B}$ creates $U_1, \ldots, U_K$ by choosing random values $\mu'_1, \mu'_2 \in \mathbb{Z}_q$ and setting:

$$U_1 := g_2^{\mu'_1 \mathbf{b}^*_1 + \mu'_2 \mathbf{f}^*_{K+1}} := g_2^{(\mu'_1 + a\mu'_2)\mathbf{b}^*_1 + \mu'_2 \mathbf{b}^*_{K+1}}.$$

In other words, $\mathcal{B}$ has implicitly set $\mu_1 := \mu'_1 + a\mu'_2$ and $\mu_2 := \mu'_2$. We note that these values are uniformly random, and μ_2 is known to $\mathcal{B}$. $\mathcal{B}$ can then form $U_2, \ldots, U_K$ as:

$$U_2 := g_2^{\mu'_1 \mathbf{b}^*_2 + \mu'_2 \mathbf{f}^*_{K+2}}, \ldots, U_K := g_2^{\mu'_1 \mathbf{b}^*_K + \mu'_2 \mathbf{f}^*_{2K}}.$$

$\mathcal{B}$ implicitly sets $\tau_1 := b, \tau_2 := c$ and computes:

$$T_1 := T^{\mathbf{f}_{K+1}} \cdot (g_1^b)^{\mathbf{f}_1}, \ldots, T_K := T^{\mathbf{f}_{2K}} \cdot (g_1^b)^{\mathbf{f}_K}.$$

If $T = g_1^{ab}$, then these are distributed as $V_1, \ldots, V_K$, since

$$T^{\mathbf{f}_{K+j}} \cdot (g_1^b)^{\mathbf{f}_j} = g_1^{\tau_1 \mathbf{b}_j}.$$

If $T = g_1^{ab+c}$, then these are distributed as $W_1, \ldots, W_K$, since

$$T^{\mathbf{f}_{k+j}} \cdot (g_1^b)^{\mathbf{f}_j} = g_1^{\tau_1 \mathbf{b}_j + \tau_2 \mathbf{b}_{K+j}}.$$

$\mathcal{B}$ then gives

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, \ldots, g_2^{\mathbf{b}_K^*}, g_2^{\mathbf{b}_{2K+1}^*}, \ldots, g_2^{\mathbf{b}_N^*}, g_1^{\mathbf{b}_1}, \ldots, g_1^{\mathbf{b}_N}, U_1, \ldots, U_K, \mu_2)$$

to $\mathcal{A}$, along with $T_1, \ldots, T_K$. $\mathcal{B}$ can then leverage $\mathcal{A}$'s advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DS1}}(\lambda)$ in distinguishing between the distributions $(V_1, \ldots, V_K)$ and $(W_1, \ldots, W_K)$ to achieve an advantage $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH1}}(\lambda)$ in distinguishing $T = g_1^{ab}$ from $T = g_1^{ab+c}$, hence violating the DDH assumption in G_1.

The dual of the Subspace assumption in G_1 is Subspace assumption in G_2 (denoted as DS2), which is identical to Definition 12 with the roles of G_1 and G_2 reversed. Similarly, we can prove that the Subspace assumption holds in G_2 if the DDH assumption in G_2 holds.

3.2 Generalized Decisional Subspace Assumption

We generalize the Decisional Subspace Assumption for Multiple Tuple of Dual Orthonormal Bases.

Definition 13 (GDS1: Generalized Decisional Subspace Assumption in G_1). *Given a group generator $\mathcal{G}(\cdot)$, define the following distribution:*

$$\begin{aligned}
&\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathrm{R}} \mathcal{G}(1^\lambda),\\
&(\mathbb{B}, \mathbb{B}^*) \leftarrow_{\mathrm{R}} \mathsf{Dual}(\mathbb{Z}_q^{N_1}, \ldots, \mathbb{Z}_q^{N_d}); \tau_1, \tau_2, \mu_1, \mu_2 \leftarrow_{\mathrm{R}} \mathbb{Z}_q,\\
&\left\{U_{1,i} := g_2^{\mu_1 \mathbf{b}_{1,i}^* + \mu_2 \mathbf{b}_{K_i+1,i}^*}, \ldots, U_{K_i,i} := g_2^{\mu_1 \mathbf{b}_{K_i,i}^* + \mu_2 \mathbf{b}_{2K_i,i}^*}\right\}_{i \in [d]},\\
&\left\{V_{1,i} := g_1^{\tau_1 \mathbf{b}_{1,i}}, \ldots, V_{K_i,i} := g_1^{\tau_1 \mathbf{b}_{K_i,i}}\right\}_{i \in [d]},\\
&\left\{W_{1,i} := g_1^{\tau_1 \mathbf{b}_{1,i} + \tau_2 \mathbf{b}_{K_i+1,i}}, \ldots, W_{K_i,i} := g_1^{\tau_1 \mathbf{b}_{K_i,i} + \tau_2 \mathbf{b}_{2K_i,i}}\right\}_{i \in [d]},\\
&D := \left(\mathbb{G}; \left\{g_2^{\mathbf{b}_{1,i}^*}, \ldots, g_2^{\mathbf{b}_{K_i,i}^*}, g_2^{\mathbf{b}_{2K_i+1,i}^*}, \ldots, g_2^{\mathbf{b}_{N_i,i}^*}, g_1^{\mathbf{b}_{1,i}}, \ldots, g_1^{\mathbf{b}_{N_i,i}}, U_{1,i}, \ldots, U_{K_i,i}\right\}_{i \in [d]}, \mu_2\right)
\end{aligned}$$

where K_i, N_i are fixed positive integers that satisfy $2K_i \leq N_i$ for $i \in [d]$. We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0,1\}$),

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{GDS1}}(\lambda) := |\Pr[\mathcal{A}(D, \{V_{1,i}, \ldots, V_{K_i}\}_{i \in [d]}) = 1] - \Pr[\mathcal{A}(D, \{W_{1,i}, \ldots, W_{K_i,i}\}_{i \in [d]}) = 1]|$$

is negligible in the security parameter λ.

Lemma 3. *If the DDH assumption in G_1 holds, then the Generalized Subspace assumption in G_1 stated in Definition 13 also holds. More precisely, for any adversary $\mathcal{A}$ against the Generalized Subspace assumption in G_1, there exist probabilistic algorithms $\mathcal{B}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{GDS1}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH1}}_{\mathcal{B}}(\lambda).$$

The proof for above lemma is essentially the same as those of Lemma 2. The dual of the Generalized Subspace assumption in G_1 is Generalized Subspace assumption in G_2 (denoted as GDS2), which is identical to Definition 13 with the roles of G_1 and G_2 reversed. Similarly, we can prove that the Generalized Subspace assumption holds in G_2 if the DDH assumption in G_2 holds.

4 Identity-Based Encryption

We first present our IBE construction along with our proof of its security under the SXDH assumption.

Construction. We begin with our IBE scheme:

- $\mathsf{Setup}(1^\lambda)$ This algorithm takes in the security parameter λ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q. The algorithm samples random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathrm{R}} \mathsf{Dual}(\mathbb{Z}_q^4)$. Let $\mathbf{d}_1, \ldots, \mathbf{d}_4$ denote the elements of $\mathbb{D}$ and $\mathbf{d}_1^*, \ldots, \mathbf{d}_4^*$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$, and outputs the public parameters as

$$\mathsf{PP} := \left\{ \mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2} \right\} \in G_T \times G_1^4 \times G_1^4$$

  and the master key

$$\mathsf{MK} := \left\{ \alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*} \right\} \in \mathbb{Z}_q \times G_2^4 \times G_2^4$$

- $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \mathrm{id})$ This algorithm picks $r \leftarrow_{\mathrm{R}} \mathbb{Z}_q$. The secret key is computed as

$$\mathsf{SK}_{\mathrm{id}} := g_2^{\alpha \mathbf{d}_1^* + r(\mathrm{id}\mathbf{d}_1^* - \mathbf{d}_2^*)} \in G_2^4.$$

- $\mathsf{Enc}(\mathsf{PP}, \mathrm{id}, \mathrm{m})$ This algorithm picks $z \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\mathsf{CT}_{\mathrm{id}} := \left\{ \mathsf{C} := \mathrm{m} \cdot (g_T^\alpha)^z, \mathsf{C}_0 := g_1^{z(\mathbf{d}_1 + \mathrm{id}\mathbf{d}_2)} \right\} \in G_T \times G_1^4.$$

- $\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathrm{id}}, \mathsf{CT}_{\mathrm{id}})$ This algorithm computes the message as

$$\mathrm{m} := \mathsf{C} / e(\mathsf{C}_0, \mathsf{SK}_{\mathrm{id}}) \in G_T.$$

We note that applying Naor's transform [9, 11] to our scheme, we can also obtain an efficient signature scheme.

Correctness. Correctness is straight-forward:

$$\begin{aligned} e(C_0, \mathsf{SK}_{\mathrm{id}}) &= e(g_1^{z(\mathbf{d}_1 + \mathrm{id}\mathbf{d}_2)}, g_2^{\alpha \mathbf{d}_1^* + r(\mathrm{id}\mathbf{d}_1^* - \mathbf{d}_2^*)}) \\ &= e(g_1, g_2)^{\alpha z \mathbf{d}_1 \cdot \mathbf{d}_1^*} \cdot e(g_1, g_2)^{zr\mathrm{id}\mathbf{d}_1 \cdot \mathbf{d}_1^* - zr\mathrm{id}\mathbf{d}_2 \cdot \mathbf{d}_2^*} \\ &= g_T^{\alpha z}. \end{aligned}$$

Proof of Security. We prove the following theorem by showing a series of lemmas.

Theorem 1. *The IBE scheme is fully secure and weakly attribute-hiding (anonymous) under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the IBE scheme, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{q_n}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH1}}_{\mathcal{B}_0}(\lambda) + \sum_{\kappa=1}^{q_n} \mathsf{Adv}^{\mathsf{DDH2}}_{\mathcal{B}_\kappa}(\lambda) + (6q_n + 3)/q$$

where q_n is the maximum number of $\mathcal{A}$'s key queries.

We adopt the dual system encryption methodology by Waters [39] to prove the security of our IBE scheme. We use the concepts of *semi-functional ciphertexts* and *semi-functional keys* in our proof and provide algorithms that generate them. We note that these algorithms are only provided for definitional purposes, and are not part of the IBE system. In particular, they do not need to be efficiently computable from the public parameters and the master key.

KeyGenSF The algorithm picks $r, \nu_1, \nu_2 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms a semi-functional secret key as

$$\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}} := g_2^{\alpha \mathbf{d}^*_1 + r(\mathrm{id}\mathbf{d}^*_1 - \mathbf{d}^*_2) + [\nu_1 \mathbf{d}^*_3 + \nu_2 \mathbf{d}^*_4]}. \tag{2}$$

EncryptSF The algorithm picks $z, \chi_1, \chi_2 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms a semi-functional ciphertext as

$$\mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}} := \left\{ \mathsf{C} := \mathrm{m} \cdot (g_T^{\alpha})^z, \mathsf{C}_0 := g_1^{z(\mathbf{d}_1 + \mathrm{id}\mathbf{d}_2) + [\chi_1 \mathbf{d}_3 + \chi_2 \mathbf{d}_4]} \right\}. \tag{3}$$

We observe that if one applies the decryption procedure with a semi-functional key and a normal ciphertext, decryption will succeed because $\mathbf{d}^*_3, \mathbf{d}^*_4$ are orthogonal to all of the vectors in exponent of C_0, and hence have no effect on decryption. Similarly, decryption of a semi-functional ciphertext by a normal key will also succeed because $\mathbf{d}_3, \mathbf{d}_4$ are orthogonal to all of the vectors in the exponent of the key. When both the ciphertext and key are semi-functional, the result of $e(\mathsf{C}_0, \mathsf{SK}_{\mathbf{v}})$ will have an additional term, namely

$$e(g_1, g_2)^{\nu_1 \chi_1 \mathbf{d}^*_3 \cdot \mathbf{d}_3 + \nu_2 \chi_2 \mathbf{d}^*_4 \cdot \mathbf{d}_4} = g_T^{(\nu_1 \chi_1 + \nu_2 \chi_2)}.$$

Decryption will then fail unless $\nu_1 \chi_1 + \nu_2 \chi_2 \equiv 0 \bmod q$. If this modular equation holds, we say that the key and ciphertext pair is *nominally semi-functional*.

For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q_n key queries $\mathbf{v}_1, \ldots, \mathbf{v}_{q_n}$, our proof of security consists of the following sequence of games between $\mathcal{A}$ and a challenger $\mathcal{B}$.

- $\mathsf{Game}_{\mathsf{Real}}$: is the real security game.
- Game_0: is the same as $\mathsf{Game}_{\mathsf{Real}}$ except that the challenge ciphertext is semi-functional.
- Game_κ: for κ from 1 to q_n, Game_κ is the same as Game_0 except that the first κ keys are semi-functional and the remaining keys are normal.
- $\mathsf{Game}_{\mathsf{Final}}$: is the same as Game_{q_n}, except that the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$. We denote the challenge ciphertext in $\mathsf{Game}_{\mathsf{Final}}$ as $\mathsf{CT}^{(\mathsf{R})}_{\mathrm{id}_{\mathrm{R}}}$.

We prove following lemmas to show the above games are indistinguishable by following an analogous strategy of [28, 30]. Our main arguments are computational indistinguishability (guaranteed by the Subspace assumptions, which are implied by the SXDH assumption) and statistical indistinguishability. The advantage gap between $\mathsf{Game}_{\mathsf{Real}}$ and Game_0 is bounded by the advantage of the Subspace assumption in G_1. Additionally, we require a statistical indistinguishability argument to show that the distribution of the challenge ciphertext remains the same from the adversary's view. For κ from 1 to q_n, the advantage gap between $\mathsf{Game}_{\kappa-1}$ and Game_κ is bounded by the advantage of Subspace assumption in G_2. Similarly, we require a statistical indistinguishability argument to show that the distribution of the the κ-th semi-functional key remains the same from the adversary's view. Finally, we statistically transform Game_{q_n} to $\mathsf{Game}_{\mathsf{Final}}$ in one step, i.e., we show the joint distributions of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathrm{id}^*_\beta}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\right\}_{\ell=1,\ldots,q_n}\right) \quad \text{and} \quad \left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\mathrm{id}_\mathsf{R}}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\right\}_{\ell=1,\ldots,q_n}\right)$$

are equivalent for the adversary's view.

We let $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Real}}}_{\mathcal{A}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 4. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Real}}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{B}_0}(\lambda) = \epsilon - 2/q$, with $K = 2$ and $N = 4$.*

Proof. $\mathcal{B}_0$ is given

$$D := \left(\mathbb{G}; g_2^{\mathbf{b}^*_1}, g_2^{\mathbf{b}^*_2}, g_1^{\mathbf{b}_1}, \ldots, g_1^{\mathbf{b}_4}, U_1, U_2, \mu_2\right)$$

along with T_1, T_2. We require that $\mathcal{B}_0$ decides whether T_1, T_2 are distributed as

$$g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2} \quad \text{or} \quad g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_3}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_4}.$$

$\mathcal{B}_0$ simulates $\mathsf{Game}_{\mathsf{Real}}$ or Game_0 with $\mathcal{A}$, depending on the distribution of T_1, T_2. To compute the public parameters and master secret key, $\mathcal{B}_0$ first chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{2\times 2}$. We implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned}
&\mathbf{d}_1 := \mathbf{b}_1, \mathbf{d}_2 := \mathbf{b}_2, \quad (\mathbf{d}_3, \ldots, \mathbf{d}_4) := (\mathbf{b}_3, \mathbf{b}_4)\mathbf{A}, \\
&\mathbf{d}^*_1 := \mathbf{b}^*_1, \mathbf{d}^*_2 := \mathbf{b}^*_2, \quad (\mathbf{d}^*_3, \ldots, \mathbf{d}^*_4) := (\mathbf{b}^*_3, \mathbf{b}^*_4)(\mathbf{A}^{-1})^\top.
\end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$. Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}^*_3}, g_2^{\mathbf{d}^*_4}$, but these will not be needed for creating normal keys. $\mathcal{B}_0$ chooses random value $\alpha \in \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}^*_1}$. It then gives $\mathcal{A}$ the public parameters

$$\mathsf{PP} := \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\right\}.$$

The master key

$$\mathsf{MK} := \left\{\alpha, g_2^{\mathbf{d}^*_1}, g_2^{\mathbf{d}^*_2}\right\}$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_0$ two pairs $(\mathrm{m}_0, \mathrm{id}_0^*)$ and $(\mathrm{m}_1, \mathrm{id}_1^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0,1\}$ and encrypts m_β under id_β^* as follows:

$$\mathsf{C} := \mathrm{m}_\beta \cdot \left(e(T_1, g_2^{\mathbf{b}_1^*})\right)^\alpha = \mathrm{m}_\beta \cdot (g_T^\alpha)^z, \quad \mathsf{C}_0 := T_1 \cdot T_2^{\mathrm{id}_\beta^*},$$

where $\mathcal{B}_0$ has implicitly set $z := \tau_1$. It gives the ciphertext $(\mathsf{C}, \mathsf{C}_0)$ to $\mathcal{A}$.

Now, if T_1, T_2 are equal to $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}$, then this is a properly distributed normal encryption of m_β. In this case, $\mathcal{B}_0$ has properly simulated $\mathsf{Game}_{\mathsf{Real}}$. If T_1, T_2 are equal to $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_3}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_4}$ instead, then the ciphertext element C_0 has an additional term of

$$\tau_2(\mathbf{b}_3 + \mathrm{id}_\beta^* \mathbf{b}_4)$$

in its exponent. The coefficients here in the basis $\mathbf{b}_3, \mathbf{b}_4$ form the vector $\tau_2(1, \mathrm{id}_\beta^*)$. To compute the coefficients in the basis $\mathbf{d}_3, \mathbf{d}_4$, we multiply the matrix $\mathbf{A}^{-1}$ by the transpose of this vector, obtaining $\tau_2 \mathbf{A}^{-1}(1, \mathrm{id}_\beta^*)^\top$. Since $\mathbf{A}$ is random (everything else given to $\mathcal{A}$ has been distributed independently of $\mathbf{A}$), these coefficients are uniformly random except with probability $2/q$ (namely, the cases τ_2 defined in Subspace problem is zero, (χ_3, χ_4) defined in Equation 3 is the zero vector) from Lemma 1 Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0. This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\mathsf{Game}_{\mathsf{Real}}$ and Game_0 to achieve an advantage $\epsilon - \frac{2}{q}$ against the Subspace assumption in G_1, namely $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{B}_0}(\lambda) = \epsilon - \frac{2}{q}$. □

Lemma 5. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\kappa-1}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_\kappa}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_\kappa$ such that $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_\kappa}(\lambda) = \epsilon - 6/q$, with $K = 2$ and $N = 4$.*

Proof. $\mathcal{B}_\kappa$ is given

$$D := \left(\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_2^{\mathbf{b}_1^*}, \ldots, g_2^{\mathbf{b}_4^*}, U_1, U_2, \mu_2\right)$$

along with T_1, T_2. We require that $\mathcal{B}_\kappa$ decides whether T_1, T_2 are distributed as

$$g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*} \quad \text{or} \quad g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_3^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_4^*}.$$

$\mathcal{B}_\kappa$ simulates Game_κ or $\mathsf{Game}_{\kappa-1}$ with $\mathcal{A}$, depending on the distribution of T_1, T_2. To compute the public parameters and master secret key, $\mathcal{B}_\kappa$ chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{2\times 2}$. We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} &\mathbf{d}_1 := \mathbf{b}_1, \mathbf{d}_2 := \mathbf{b}_2, \quad (\mathbf{d}_3, \mathbf{d}_4) := (\mathbf{b}_3, \mathbf{b}_4)\mathbf{A}, \\ &\mathbf{d}_1^* := \mathbf{b}_1^*, \mathbf{d}_2^* := \mathbf{b}_2^*, \quad (\mathbf{d}_3^*, \mathbf{d}_4^*) := (\mathbf{b}_3^*, \mathbf{b}_4^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$. $\mathcal{B}_\kappa$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. $\mathcal{B}$ can gives $\mathcal{A}$ the public parameters

$$\mathsf{PP} := \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\right\}.$$

The master key

$$\mathsf{MK} := \left\{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}\right\}$$

is known to $\mathcal{B}_\kappa$, which allows $\mathcal{B}_\kappa$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm. Since $\mathcal{B}_\kappa$ also knows $g_2^{\mathbf{d}_3^*}, g_2^{\mathbf{d}_4^*}$, it can easily produce semi-functional keys. To answer the first $\kappa - 1$ key queries that $\mathcal{A}$ makes, $\mathcal{B}_\kappa$ runs the semi-functional key generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ-th key query for id_κ, $\mathcal{B}_\kappa$ responds with:

$$\mathsf{SK}_{\mathrm{id}_\kappa} := (g_2^{\mathbf{b}_1^*})^\alpha \cdot T_1^{\mathrm{id}_\kappa} \cdot T_2^{-1}.$$

This implicitly sets $r := \tau_1$. If T_1, T_2 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}$, then this is a properly distributed normal key. If T_1, T_2 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_3^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_4^*}$, then this is a semi-functional key, whose exponent vector includes

$$\tau_2(\mathrm{id}_\kappa \mathbf{b}_3^* - \mathbf{b}_4^*) \tag{4}$$

as its component in the span of $\mathbf{b}_3^*, \mathbf{b}_4^*$. To respond to the remaining key queries, $\mathcal{B}_\kappa$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_\kappa$ two pairs $(\mathrm{m}_0, \mathrm{id}_0^*)$ and $(\mathrm{m}_1, \mathrm{id}_1^*)$. $\mathcal{B}_\kappa$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts m_β under id_β^* as follows:

$$\mathsf{C} := \mathrm{m}_\beta \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \mathrm{m}_\beta \cdot (g_T^\alpha)^z, \quad \mathsf{C}_0 := U_1 \cdot U_2^{\mathrm{id}_\beta^*},$$

where $\mathcal{B}_\kappa$ has implicitly set $z := \mu_1$. The "semi-functional part" of the exponent vector here is:

$$\mu_2(\mathbf{b}_3 + \mathrm{id}_\beta^* \mathbf{b}_4). \tag{5}$$

We observe that if $\mathrm{id}_\beta^* = \mathrm{id}_\kappa$ (which is not allowed), then vectors in Equations 4 and 5 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair. It gives the ciphertext $(\mathsf{C}, \mathsf{C}_0)$ to $\mathcal{A}$.

We now argue that since $\mathrm{id}_\beta^* \neq \mathrm{id}_\kappa$, in $\mathcal{A}$'s view the vectors in Equations 4 and 5 are distributed as random vectors in the spans of $\mathbf{d}_3^*, \mathbf{d}_4^*$ and $\mathbf{d}_3, \mathbf{d}_4$ respectively. To see this, we take the coefficients of vectors in Equations 4 and 5 in terms of the bases $\mathbf{b}_3^*, \mathbf{b}_4^*$ and $\mathbf{b}_3, \mathbf{b}_4$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_3^*, \mathbf{d}_4^*$ and $\mathbf{d}_3, \mathbf{d}_4$. Using the change of basis matrix $\mathbf{A}$, we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\mathrm{id}_\kappa, -1)^\top,\ \mu_2 \mathbf{A}^{-1} (1, \mathrm{id}_\beta^*).$$

Since the distribution of everything given to $\mathcal{A}$ except for the κ-th key and the challenge ciphertext is independent of the random matrix $\mathbf{A}$ and $\mathrm{id}_\beta^* \neq \mathrm{id}_\kappa$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_1, χ_2) or (ν_1, ν_2) defined in Equations 3 and 2 is the zero vector) from Lemma 1. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

If T_1, T_2 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}$, then the coefficients of the vector in Equation 5 are uniformly except with probability $2/q$ (namely, the cases $\mu_2 = 1$ defined in Subspace problem is zero, (χ_1, χ_2) defined in Equation 3 is the zero vector) from Lemma 1. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

In summary, $\mathcal{B}_\kappa$ has properly simulated either $\mathsf{Game}_{\kappa-1}$ or Game_κ for $\mathcal{A}$, depending on the distribution of T_1, T_2. It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - 6/q$ against the Subspace assumption in G_2, namely $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_\kappa}(\lambda) = \epsilon - 6/q$. □

Lemma 6. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_{q_n}}_{\mathcal{A}}(\lambda) + 1/q$.*

Proof. To prove this lemma, we show the joint distributions of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathrm{id}^*_\beta}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\right\}_{\ell\in[q_n]}\right)$$

in Game_{q_n} and that of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\mathrm{id}_\mathsf{R}}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\right\}_{\ell\in[q_n]}\right)$$

in $\mathsf{Game}_{\mathsf{Final}}$ are equivalent for the adversary's view, where $\mathsf{CT}^{(\mathsf{R})}_{\mathrm{id}_\mathsf{R}}$ is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$.

For this purpose, we pick $\mathbf{A} := (\xi_{i,j}) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{2\times 2}$ and define new dual orthonormal bases $\mathbb{F} := (\mathbf{f}_1, \ldots, \mathbf{f}_4)$, and $\mathbb{F}^* := (\mathbf{f}^*_1, \ldots, \mathbf{f}^*_4)$ as follows:

$$\begin{pmatrix}\mathbf{f}_1\\ \mathbf{f}_2\\ \mathbf{f}_3\\ \mathbf{f}_4\end{pmatrix} := \begin{pmatrix}1 & 0 & 0\,0\\ 0 & 1 & 0\,0\\ \xi_{1,1} & \xi_{1,2} & 1\,0\\ \xi_{2,1} & \xi_{2,2} & 0\,1\end{pmatrix}\begin{pmatrix}\mathbf{d}_1\\ \mathbf{d}_2\\ \mathbf{d}_3\\ \mathbf{d}_4\end{pmatrix}, \quad \begin{pmatrix}\mathbf{f}^*_1\\ \mathbf{f}^*_2\\ \mathbf{f}^*_3\\ \mathbf{f}^*_4\end{pmatrix} := \begin{pmatrix}1\,0 & -\xi_{1,1} & -\xi_{2,1}\\ 0\,1 & -\xi_{1,2} & -\xi_{2,2}\\ 0\,0 & 1 & 0\\ 0\,0 & 0 & 1\end{pmatrix}\begin{pmatrix}\mathbf{d}^*_1\\ \mathbf{d}^*_2\\ \mathbf{d}^*_3\\ \mathbf{d}^*_4\end{pmatrix}.$$

It is easy to verify that $\mathbb{F}$ and $\mathbb{F}^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}$ and $\mathbb{D}^*$.

Then the public parameters, challenge ciphertext, and queried secret keys, $(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathrm{id}^*_\beta}, \{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\}_{\ell\in[q_n]})$ in Game_{q_n} are expressed over bases $\mathbb{D}$ and $\mathbb{D}^*$ as

$$\begin{aligned}
&\mathsf{PP} := \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\right\},\\
&\mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta} := \left\{\mathsf{C} := \mathrm{m}\cdot(g_T^\alpha)^z, \quad \mathsf{C}_0 := g_1^{z(\mathbf{d}_1 + \mathrm{id}^*_\beta\mathbf{d}_2) + [\chi_1\mathbf{d}_3 + \chi_2\mathbf{d}_4]}\right\},\\
&\left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell} := g_2^{\alpha\mathbf{d}^*_1 + r_\ell(\mathrm{id}_\ell\mathbf{d}^*_1 - \mathbf{d}^*_2) + [\nu_{1,\ell}\mathbf{d}^*_3 + \nu_{2,\ell}\mathbf{d}^*_4]}\right\}_{\ell\in[q_n]}.
\end{aligned}$$

Then we can express them over bases $\mathbb{F}$ and $\mathbb{F}^*$ as

$$\begin{aligned}
&\mathsf{PP} := \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{f}_1}, g_1^{\mathbf{f}_2}\right\},\\
&\mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta} := \left\{\mathsf{C} := \mathrm{m}\cdot(g_T^\alpha)^z, \quad \mathsf{C}_0 := g_1^{(z'_1\mathbf{f}_1 + z'_2\mathbf{f}_2) + [\chi_1\mathbf{d}_3 + \chi_2\mathbf{d}_4]}\right\},\\
&\left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell} := g_2^{\alpha\mathbf{f}^*_1 + r_\ell(\mathrm{id}_\ell\mathbf{f}^*_1 - \mathbf{f}^*_2) + [\nu'_{1,\ell}\mathbf{f}^*_3 + \nu'_{2,\ell}\mathbf{f}^*_4]}\right\}_{\ell\in[q_n]},
\end{aligned}$$

where

$$\begin{aligned}
&z'_1 := z - \chi_1\xi_{1,1} - \chi_2\xi_{2,1},\\
&z'_2 := z\mathrm{id}^*_\beta - \chi_1\xi_{1,2} - \chi_2\xi_{2,2},\\
&\left\{\begin{array}{l}\nu'_{1,\ell} := \nu_{1,\ell} + \alpha\xi_{1,1} + r_\ell(\mathrm{id}_\ell\xi_{1,1} - \xi_{1,2})\\ \nu'_{2,\ell} := \nu_{2,\ell} + \alpha\xi_{1,2} + r_\ell(\mathrm{id}_\ell\xi_{2,1} - \xi_{2,2})\end{array}\right\}_{\ell\in[q_n]},
\end{aligned}$$

which are all uniformly distributed if (χ_1, χ_2) defined in Equation 3 is a non-zero vector since $z, \{\xi_{i,j}\}_{i\in[d],j\in[2]}, \{\nu_{1,\ell}, \nu_{2,\ell}\}_{\ell\in[q_n]}$ are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients $s(1, \mathrm{id}^*_\beta)$ of $\mathbf{d}_1, \mathbf{d}_2$ in the C_1 term of the challenge ciphertext is changed to random coefficients $(z'_1, z'_2) \in \mathbb{Z}_q^n$ of $\mathbf{f}_1, \mathbf{f}_2$, thus the challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$. Moreover, all coefficients $\{(\nu'_{1,\ell}, \nu'_{2,\ell})\}_{\ell\in[q_n]}$ of $\mathbf{f}^*_3, \mathbf{f}^*_4$ in the $\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\}_{\ell\in[q_n]}$ are all uniformly distributed since $\{(\nu_{1,\ell}, \nu_{2,\ell})\}_{\ell\in[q_n]}$ of $\mathbf{d}^*_3, \mathbf{d}^*_4$ are all independent random values. Thus

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathrm{id}^*_\beta}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\right\}_{\ell\in[q_n]}\right)$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\mathrm{id}_\mathsf{R}}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathrm{id}_\ell}\right\}_{\ell\in[q_n]}\right)$$

in $\mathsf{Game}_{\mathsf{Final}}$.

In the adversary's view, both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same public parameters. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{q_n} over bases $(\mathbb{D}, \mathbb{D}^*)$ and in $\mathsf{Game}_{\mathsf{Final}}$ over bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, Game_{q_n} and $\mathsf{Game}_{\mathsf{Final}}$ are statistically indistinguishable except with probability $1/q$ (namely, the case $(\chi_1, \chi_2) = \mathbf{0}$). □

Lemma 7. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in $\mathsf{Game}_{\mathsf{Final}}$. Hence, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) = 0$. □

In $\mathsf{Game}_{\mathsf{Final}}$, the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$, independent of the two messages and the challenge identities provided by $\mathcal{A}$. Thus, our IBE scheme is weakly attribute-hiding (anonymous).

5 A Signature Scheme

In this section, we present the signature scheme derived from the preceding IBE scheme via Naor's transform. The security of the signature scheme follows from the full security of our IBE scheme.

- $\mathsf{KeyGen}(1^\lambda)$ This algorithm takes in the security parameter λ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q. The algorithm samples random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \xleftarrow{R} \mathsf{Dual}(\mathbb{Z}_q^4)$. Let $\mathbf{d}_1, \ldots, \mathbf{d}_4$ denote the elements of $\mathbb{D}$ and $\mathbf{d}^*_1, \ldots, \mathbf{d}^*_4$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}^*_1}$, and outputs the public key as
$$\mathsf{PK} = \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\},$$
and the signing key
$$\mathsf{SK} = \{\alpha, g_2^{\mathbf{d}^*_1}, g_2^{\mathbf{d}^*_2}\}.$$

– $\mathsf{Sign}(\mathsf{PK}, \mathsf{SK}, \mathrm{m})$ This algorithm picks $r \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and computes the signature as

$$\sigma = g_2^{(\alpha + r\mathrm{m})\mathbf{d}_1^* - r\mathbf{d}_2^*}.$$

– $\mathsf{Verify}(\mathsf{PK}, \sigma, \mathrm{m})$ This algorithm verifies a signature σ by testing whether $e(g_1^{\mathbf{d}_1 + \mathrm{m}\mathbf{d}_2}, \sigma) = e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \boldsymbol{d}_1^*}$.[3] If the equality holds the signature is declared `valid`; otherwise it is declared `invalid`.

6 Inner Product Encryption

We now present our IPE scheme, the construction and security proof of which are essentially the same as our IBE except that we extend the embedded equality relation to general inner product relation.

Construction. We begin with our IPE scheme:

– $\mathsf{Setup}(1^\lambda)$ This algorithm takes in the security parameter λ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q. The algorithm samples random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathrm{R}} \mathsf{Dual}(\mathbb{Z}_q^{2n})$. Let $\mathbf{d}_1, \ldots, \mathbf{d}_{2n}$ denote the elements of $\mathbb{D}$ and $\mathbf{d}_1^*, \ldots, \mathbf{d}_{2n}^*$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, computes $g_T := e(g_1, g_2)^{\mathbf{d}_1 \cdot \mathbf{d}_1^*}$, and outputs the public parameters as

$$\mathsf{PP} := \left\{ \mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \ldots, g_1^{\mathbf{d}_n} \right\} \in G_T \times (G_1^{2n})^n$$

and the master key

$$\mathsf{MK} := \left\{ \alpha, g_2^{\mathbf{d}_1^*}, \ldots, g_2^{\mathbf{d}_n^*} \right\} \in \mathbb{Z}_q \times (G_2^{2n})^n.$$

– $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \mathbf{v} := (v_1, \ldots, v_n))$ This algorithm picks $r \leftarrow_{\mathrm{R}} \mathbb{Z}_q$. The secret key is computed as

$$\mathsf{SK}_{\mathbf{v}} := g_2^{\alpha \mathbf{d}_1^* + r(v_1 \mathbf{d}_1^* + \cdots + v_n \mathbf{d}_n^*)} \in G_2^{2n}.$$

– $\mathsf{Enc}(\mathsf{PP}, \mathbf{x} := (x_1, \ldots, x_n), \mathrm{m})$ WLOG, we assume that $x_1 = 1$. This algorithm picks $z \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\mathsf{CT}_{\mathbf{x}} := \left\{ \mathsf{C} := \mathrm{m} \cdot (g_T^\alpha)^z, \mathsf{C}_0 := g_1^{z(x_1 \mathbf{d}_1 + \cdots + x_n \mathbf{d}_n)} \right\} \in G_T \times G_1^{2n}.$$

– $\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathbf{v}}, \mathsf{CT}_{\mathbf{x}})$ This algorithm computes the message as

$$\mathrm{m} := \mathsf{C} / e(\mathsf{C}_0, \mathsf{SK}_{\mathbf{v}}) \in G_T.$$

Correctness. Correctness is straight-forward:

$$\begin{aligned} e(C_0, \mathsf{SK}_{\mathbf{v}}) &= e(g_1^{z(x_1 \mathbf{d}_1 + \cdots + x_n \mathbf{d}_n)}, g_2^{\alpha \mathbf{d}_1^* + r(v_1 \mathbf{d}_1^* + \cdots + v_n \mathbf{d}_n^*)}) \\ &= e(g_1, g_2)^{\alpha z x_1 \mathbf{d}_1 \cdot \mathbf{d}_1^*} \cdot e(g_1, g_2)^{zr(\underline{v_1 x_1 \mathbf{d}_1 \cdot \mathbf{d}_1^* + \cdots + v_n x_n \mathbf{d}_n \cdot \mathbf{d}_n^*})} \\ &= g_T^{\alpha z} \cdot g_T^{zr\underline{\mathbf{v} \cdot \mathbf{x}}} \\ &= g_T^{\alpha z}. \end{aligned}$$

[3] Directly applying Naor's transform yields a verification algorithm that works as follows: pick $z \leftarrow \mathbb{Z}_q$, and test whether $e(g_1^{(\mathbf{d}_1 + \mathrm{m}\mathbf{d}_2)z}, \sigma) = (e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*})^z$. With overwhelming probability over z, this agrees with the verification algorithm as written.

Proof of Security. We prove the following theorem by showing a series of lemmas.

Theorem 2. *The IPE scheme is fully secure and weakly attribute-hiding under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the IPE scheme, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{q_n}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{IPE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH1}}_{\mathcal{B}_0}(\lambda) + \sum_{\kappa=1}^{q_n} \mathsf{Adv}^{\mathsf{DDH2}}_{\mathcal{B}_\kappa}(\lambda) + (6q_n + 3)/q$$

where q_n is the maximum number of $\mathcal{A}$'s key queries.

We adopt the dual system encryption methodology by Waters [39] to prove the security of our IPE scheme, the strategy is essentially the same as our IBE scheme. We first define *semi-functional ciphertexts* and *semi-functional keys* in our proof and provide algorithms that generate them.

KeyGenSF The algorithm picks $r, \nu_1, \ldots, \nu_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms a semi-functional secret key as

$$\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}} := g_2^{\alpha \mathbf{d}^*_1 + r(v_1 \mathbf{d}^*_1 + \ldots + v_n \mathbf{d}^*_n) + [\nu_1 \mathbf{d}^*_{n+1} + \cdots + \nu_n \mathbf{d}^*_{2n}]}. \tag{6}$$

EncryptSF The algorithm picks $z, \chi_1, \ldots, \chi_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms a semi-functional ciphertext as

$$\mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}} := \left\{ \mathsf{C} := \mathrm{m} \cdot (g_T^{\alpha})^z, \mathsf{C}_0 := g_1^{z(x_1 \mathbf{d}_1 + \ldots + x_n \mathbf{d}_n) + [\chi_1 \mathbf{d}_{n+1} + \cdots + \chi_n \mathbf{d}_{2n}]} \right\}. \tag{7}$$

We observe that if one applies the decryption procedure with a semi-functional key and a normal ciphertext, decryption will succeed because $\mathbf{d}^*_{n+1}, \ldots, \mathbf{d}^*_{2n}$ are orthogonal to all of the vectors in exponent of C_0, and hence have no effect on decryption. Similarly, decryption of a semi-functional ciphertext by a normal key will also succeed because $\mathbf{d}_{n+1}, \ldots, \mathbf{d}_{2n}$ are orthogonal to all of the vectors in the exponent of the key. When both the ciphertext and key are semi-functional, the result of $e(\mathsf{C}_0, \mathsf{SK}_{\mathbf{v}})$ will have an additional term, namely

$$e(g_1, g_2)^{\nu_1 \chi_1 \mathbf{d}^*_{n+1} \cdot \mathbf{d}_{n+1} + \ldots + \nu_n \chi_n \mathbf{d}^*_{2n} \cdot \mathbf{d}_{2n}} = g_T^{(\nu_1 \chi_1 + \ldots + \nu_n \chi_n)}.$$

Decryption will then fail unless $\nu_1 \chi_1 + \ldots + \nu_n \chi_n \equiv 0 \bmod q$. If this modular equation holds, we say that the key and ciphertext pair is *nominally semi-functional*.

For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q_n key queries $\mathbf{v}_1, \ldots, \mathbf{v}_{q_n}$, our proof of security consists of the following sequence of games between $\mathcal{A}$ and a challenger $\mathcal{B}$.

- $\mathsf{Game}_{\mathsf{Real}}$: is the real security game.
- Game_0: is the same as $\mathsf{Game}_{\mathsf{Real}}$ except that the challenge ciphertext is semi-functional.
- Game_κ: for κ from 1 to q_n, Game_κ is the same as Game_0 except that the first κ keys are semi-functional and the remaining keys are normal.
- $\mathsf{Game}_{\mathsf{Final}}$: is the same as Game_{q_n}, except that the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$. We denote the challenge ciphertext in $\mathsf{Game}_{\mathsf{Final}}$ as $\mathsf{CT}^{(\mathsf{R})}_{\mathbf{x}_{\mathsf{R}}}$.

We let $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Real}}}_{\mathcal{A}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 8. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Real}}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{B}_0}(\lambda) = \epsilon - 2/q$, with $K = n$ and $N = 2n$.*

Proof. $\mathcal{B}_0$ is given

$$D := \left(\mathbb{G}; g_2^{\mathbf{b}_1^*}, \ldots, g_2^{\mathbf{b}_n^*}, g_1^{\mathbf{b}_1}, \ldots, g_1^{\mathbf{b}_{2n}}, U_1, \ldots, U_n, \mu_2\right)$$

along with $T_1, \ldots, T_n$. We require that $\mathcal{B}_0$ decides whether $T_1, \ldots, T_n$ are distributed as

$$g_1^{\tau_1 \mathbf{b}_1}, \ldots, g_1^{\tau_1 \mathbf{b}_n} \quad \text{or} \quad g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{n+1}}, \ldots, g_1^{\tau_1 \mathbf{b}_n + \tau_2 \mathbf{b}_{2n}}.$$

$\mathcal{B}_0$ simulates $\mathsf{Game}_{\mathsf{Real}}$ or Game_0 with $\mathcal{A}$, depending on the distribution of $T_1, \ldots, T_n$. To compute the public parameters and master secret key, $\mathcal{B}_0$ first chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$. We implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned}
\mathbf{d}_1 &:= \mathbf{b}_1, \ldots, \mathbf{d}_n := \mathbf{b}_n, \quad (\mathbf{d}_{n+1}, \ldots, \mathbf{d}_{2n}) := (\mathbf{b}_{n+1}, \ldots, \mathbf{b}_{2n})\mathbf{A}, \\
\mathbf{d}_1^* &:= \mathbf{b}_1^*, \ldots, \mathbf{d}_n^* := \mathbf{b}_n^*, \quad (\mathbf{d}_{n+1}^*, \ldots, \mathbf{d}_{2n}^*) := (\mathbf{b}_{n+1}^*, \ldots, \mathbf{b}_{2n}^*)(\mathbf{A}^{-1})^\top.
\end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$. Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}_{n+1}^*}, \ldots, g_2^{\mathbf{d}_{2n}^*}$, but these will not be needed for creating normal keys. $\mathcal{B}_0$ chooses random value $\alpha \in \mathbb{Z}_q$ and computes $e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. It then gives $\mathcal{A}$ the public parameters

$$\mathsf{PP} := \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \ldots, g_1^{\mathbf{d}_n}\right\}.$$

The master key

$$\mathsf{MK} := \left\{\alpha, g_2^{\mathbf{d}_1^*}, \ldots, g_2^{\mathbf{d}_n^*}\right\}$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_0$ two pairs $(\mathrm{m}_0, \mathbf{x}_0^*)$ and $(\mathrm{m}_1, \mathbf{x}_1^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts m_β under $\mathbf{x}_\beta^* := (x_{1,\beta}^*, \ldots, x_{n,\beta}^*)$ as follows:

$$\mathsf{C} := \mathrm{m}_\beta \cdot \left(e(T_1, g_2^{\mathbf{b}_1^*})\right)^\alpha = \mathrm{m}_\beta \cdot (g_T^\alpha)^z, \quad \mathsf{C}_0 := T_1^{x_{1,\beta}^*} \cdots T_n^{x_{n,\beta}^*},$$

where $\mathcal{B}_0$ has implicitly set $z := \tau_1$. It gives the ciphertext $(\mathsf{C}, \mathsf{C}_0)$ to $\mathcal{A}$.

Now, if $T_1, \ldots, T_n$ are equal to $g_1^{\tau_1 \mathbf{b}_1}, \ldots, g_1^{\tau_1 \mathbf{b}_n}$, then this is a properly distributed normal encryption of m_β. In this case, $\mathcal{B}_0$ has properly simulated $\mathsf{Game}_{\mathsf{Real}}$. If $T_1, \ldots, T_n$ are equal to $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{n+1}}, \ldots, g_1^{\tau_1 \mathbf{b}_n + \tau_2 \mathbf{b}_{2n}}$ instead, then the ciphertext element C_0 has an additional term of

$$\tau_2(x_{1,\beta}^* \mathbf{b}_{n+1} + \cdots + x_{n,\beta}^* \mathbf{b}_{2n})$$

in its exponent. The coefficients here in the basis $\mathbf{b}_{n+1}, \ldots, \mathbf{b}_{2n}$ form the vector $\tau_2(x_{1,\beta}^*, \ldots, x_{n,\beta}^*)$. To compute the coefficients in the basis $\mathbf{d}_{n+1}, \ldots, \mathbf{d}_{2n}$, we multiply the matrix $\mathbf{A}^{-1}$ by the transpose of this vector, obtaining $\tau_2 \mathbf{A}^{-1}(x_{1,\beta}^*, \ldots, x_{n,\beta}^*)^\top$. Since $\mathbf{A}$ is random (everything else given to $\mathcal{A}$ has been distributed independently of $\mathbf{A}$), these coefficients are uniformly random except with probability $2/q$ (namely, the cases τ_2 defined in Subspace problem is zero, $(\chi_1, \ldots, \chi_n)$ defined in Equation 7 is the zero vector) from Lemma 1 Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0. This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\mathsf{Game}_{\mathsf{Real}}$ and Game_0 to achieve an advantage $\epsilon - \frac{2}{q}$ against the Subspace assumption in G_1, namely $\mathsf{Adv}_{\mathcal{B}_0}^{\mathsf{DS1}}(\lambda) = \epsilon - \frac{2}{q}$. □

Lemma 9. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\kappa-1}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_{\kappa}}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_\kappa$ such that $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_\kappa}(\lambda) = \epsilon - 6/q$, with $K = n$ and $N = 2n$.*

Proof. $\mathcal{B}_\kappa$ is given

$$D := \left(\mathbb{G}; g_1^{\mathbf{b}_1}, \ldots, g_1^{\mathbf{b}_n}, g_2^{\mathbf{b}^*_1}, \ldots, g_2^{\mathbf{b}^*_{2n}}, U_1, \ldots, U_n, \mu_2\right)$$

along with $T_1, \ldots, T_n$. We require that $\mathcal{B}_\kappa$ decides whether $T_1, \ldots, T_n$ are distributed as

$$g_2^{\tau_1 \mathbf{b}^*_1}, \ldots, g_2^{\tau_1 \mathbf{b}^*_n} \quad \text{or} \quad g_2^{\tau_1 \mathbf{b}^*_1 + \tau_2 \mathbf{b}^*_{n+1}}, \ldots, g_2^{\tau_1 \mathbf{b}^*_n + \tau_2 \mathbf{b}^*_{2n}}.$$

$\mathcal{B}_\kappa$ simulates Game_κ or $\mathsf{Game}_{\kappa-1}$ with $\mathcal{A}$, depending on the distribution of $T_1, \ldots, T_n$. To compute the public parameters and master secret key, $\mathcal{B}_\kappa$ chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$. We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned}
\mathbf{d}_1 &:= \mathbf{b}_1, \ldots, \mathbf{d}_n := \mathbf{b}_n, \quad (\mathbf{d}_{n+1}, \ldots, \mathbf{d}_{2n}) := (\mathbf{b}_{n+1}, \ldots, \mathbf{b}_{2n})\mathbf{A}, \\
\mathbf{d}^*_1 &:= \mathbf{b}^*_1, \ldots, \mathbf{d}^*_n := \mathbf{b}^*_n, \quad (\mathbf{d}^*_{n+1}, \ldots, \mathbf{d}^*_{2n}) := (\mathbf{b}^*_{n+1}, \ldots, \mathbf{b}^*_{2n})(\mathbf{A}^{-1})^\top.
\end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$. $\mathcal{B}_\kappa$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}^*_1}$. $\mathcal{B}$ can gives $\mathcal{A}$ the public parameters

$$\mathsf{PP} := \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \ldots, g_1^{\mathbf{d}_n}\right\}.$$

The master key

$$\mathsf{MK} := \left\{\alpha, g_2^{\mathbf{d}^*_1}, \ldots, g_2^{\mathbf{d}^*_n}\right\}$$

is known to $\mathcal{B}_\kappa$, which allows $\mathcal{B}_\kappa$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm. Since $\mathcal{B}_\kappa$ also knows $g_2^{\mathbf{d}^*_{n+1}}, \ldots, g_2^{\mathbf{d}^*_{2n}}$, it can easily produce semi-functional keys. To answer the first $\kappa - 1$ key queries that $\mathcal{A}$ makes, $\mathcal{B}_\kappa$ runs the semi-functional key generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ-th key query for $\mathbf{v}_\kappa := (v_1, \ldots, v_n)$, $\mathcal{B}_\kappa$ responds with:

$$\mathsf{SK}_{\mathbf{v}_\kappa} := (g_2^{\mathbf{b}^*_1})^\alpha \cdot T_1^{v_1} \cdots T_n^{v_n}.$$

This implicitly sets $r := \tau_1$. If $T_1, \ldots, T_n$ are equal to $g_2^{\tau_1 \mathbf{b}^*_1}, \ldots, g_2^{\tau_1 \mathbf{b}^*_n}$, then this is a properly distributed normal key. If $T_1, \ldots, T_n$ are equal to $g_2^{\tau_1 \mathbf{b}^*_1 + \tau_2 \mathbf{b}^*_{n+1}}, \ldots, g_2^{\tau_1 \mathbf{b}^*_n + \tau_2 \mathbf{b}^*_{2n}}$, then this is a semi-functional key, whose exponent vector includes

$$\tau_2(v_1 \mathbf{b}^*_{n+1} + \cdots + v_n \mathbf{b}^*_{2n}) \tag{8}$$

as its component in the span of $\mathbf{b}^*_{n+1}, \ldots, \mathbf{b}^*_{2n}$. To respond to the remaining key queries, $\mathcal{B}_\kappa$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_\kappa$ two pairs $(\mathrm{m}_0, \mathbf{x}^*_0)$ and $(\mathrm{m}_1, \mathbf{x}^*_1)$. $\mathcal{B}_\kappa$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts m_β under $\mathbf{x}^*_\beta := (x^*_{1,\beta}, \ldots, x^*_{n,\beta})$ as follows:

$$\mathsf{C} := \mathrm{m}_\beta \cdot \left(e(U_1, g_2^{\mathbf{b}^*_1})\right)^\alpha = \mathrm{m}_\beta \cdot (g_T^\alpha)^z, \quad \mathsf{C}_0 := U_1^{x^*_{1,\beta}} \cdots U_n^{x^*_{n,\beta}},$$

where $\mathcal{B}_\kappa$ has implicitly set $z := \mu_1$. The "semi-functional part" of the exponent vector here is:

$$\mu_2(x^*_{1,\beta}\mathbf{b}_{n+1} + \ldots + x^*_{n,\beta}\mathbf{b}_{2n}). \tag{9}$$

We observe that if $\mathbf{x}^*_\beta \cdot \mathbf{v}_\kappa = 0$ (which is not allowed), then vectors 8 and 9 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair. It gives the ciphertext $(\mathsf{C}, \mathsf{C}_0)$ to $\mathcal{A}$.

We now argue that since $\mathbf{x}^*_\beta \cdot \mathbf{v}_\kappa \neq 0$, in $\mathcal{A}$'s view the vectors 8 and 9 are distributed as random vectors in the spans of $\mathbf{d}^*_{n+1}, \ldots, \mathbf{d}^*_{2n}$ and $\mathbf{d}_{n+1}, \ldots, \mathbf{d}_{2n}$ respectively. To see this, we take the coefficients of vectors 8 and 9 in terms of the bases $\mathbf{b}^*_{n+1}, \ldots, \mathbf{b}^*_{2n}$ and $\mathbf{b}_{n+1}, \ldots, \mathbf{b}_{2n}$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}^*_{n+1}, \ldots, \mathbf{d}^*_{2n}$ and $\mathbf{d}_{n+1}, \ldots, \mathbf{d}_{2n}$. Using the change of basis matrix $\mathbf{A}$, we obtain the new coefficients (in vector form) as:

$$\tau_2\mathbf{A}^\top(v_1, \ldots, v_n)^\top,\ \mu_2\mathbf{A}^{-1}(x^*_{1,\beta}, \ldots, x^*_{n,\beta}).$$

Since the distribution of everything given to $\mathbf{A}$ except for the κ-th key and the challenge ciphertext is independent of the random matrix $\mathbf{A}$ and $\mathbf{x}^*_\beta \cdot \mathbf{v}_\kappa \neq 0$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, $(\chi_1, \ldots, \chi_n)$ or $(\nu_1, \ldots, \nu_n)$ defined in Equations 7 and 6 is the zero vector) from Lemma 1. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

If $T_1, \ldots, T_n$ are equal to $g_2^{\tau_1\mathbf{b}^*_1}, \ldots, g_2^{\tau_1\mathbf{b}^*_n}$, then the coefficients of the vector 9 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, $(\chi_1, \ldots, \chi_n)$ defined in Equation 7 is the zero vector) from Lemma 1. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

In summary, $\mathcal{B}_\kappa$ has properly simulated either $\mathsf{Game}_{\kappa-1}$ or Game_κ for $\mathcal{A}$, depending on the distribution of $T_1, \ldots, T_n$. It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - 6/q$ against the Subspace assumption in G_2, namely $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_\kappa}(\lambda) = \epsilon - 6/q$. □

Lemma 10. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_{q_n}}_{\mathcal{A}}(\lambda) + 1/q$.*

Proof. To prove this lemma, we show the joint distributions of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\right\}_{\ell\in[q_n]}\right)$$

in Game_{q_n} and that of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\mathbf{x}_\mathsf{R}}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\right\}_{\ell\in[q_n]}\right)$$

in $\mathsf{Game}_{\mathsf{Final}}$ are equivalent for the adversary's view, where $\mathsf{CT}^{(\mathsf{R})}_{\mathbf{v}_\mathsf{R}}$ is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$.

For this purpose, we pick $\mathbf{A} := (\xi_{i,j}) \leftarrow_\mathrm{R} \mathbb{Z}_q^{n\times n}$ and define new dual orthonormal bases $\mathbb{F} := (\mathbf{f}_1, \ldots, \mathbf{f}_{2n})$, and $\mathbb{F}^* := (\mathbf{f}^*_1, \ldots, \mathbf{f}^*_{2n})$ as follows:

$$\begin{pmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_n\\ \mathbf{f}_{n+1}\\ \vdots\\ \mathbf{f}_{2n}\end{pmatrix} := \begin{pmatrix} 1 & \cdots & 0 & 0\cdots 0\\ \vdots & \ddots & \vdots & \vdots\ddots\vdots\\ 0 & \cdots & 1 & 0\cdots 0\\ \xi_{1,1} & \cdots & \xi_{1,n} & 1\cdots 0\\ \vdots & \ddots & \vdots & \vdots\ddots\vdots\\ \xi_{n,1} & \cdots & \xi_{n,n} & 0\cdots 1\end{pmatrix}\begin{pmatrix}\mathbf{d}_1\\ \vdots\\ \mathbf{d}_n\\ \mathbf{d}_{n+1}\\ \vdots\\ \mathbf{d}_{2n}\end{pmatrix},\quad \begin{pmatrix}\mathbf{f}^*_1\\ \vdots\\ \mathbf{f}^*_n\\ \mathbf{f}^*_{n+1}\\ \vdots\\ \mathbf{f}^*_{2n}\end{pmatrix} := \begin{pmatrix} 1\cdots 0 & -\xi_{1,1} & \cdots & -\xi_{n,1}\\ \vdots\ddots\vdots & \vdots & \ddots & \vdots\\ 0\cdots 1 & -\xi_{1,n} & \cdots & -\xi_{n,n}\\ 0\cdots 0 & 1 & \cdots & 0\\ \vdots\ddots\vdots & \vdots & \ddots & \vdots\\ 0\cdots 0 & 0 & \cdots & 1\end{pmatrix}\begin{pmatrix}\mathbf{d}^*_1\\ \vdots\\ \mathbf{d}^*_n\\ \mathbf{d}^*_{n+1}\\ \vdots\\ \mathbf{d}^*_{2n}\end{pmatrix}.$$

It is easy to verify that $\mathbb{F}$ and $\mathbb{F}^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}$ and $\mathbb{D}^*$.

Then the public parameters, challenge ciphertext for $\mathbf{x}^*_\beta := (x^*_{1,\beta}, \ldots, x^*_{n,\beta})$, and queried secret keys for $\{\mathbf{v}_\ell := (v_{1,\ell}, \ldots, v_{n,\ell})\}_{\ell\in[q_n]}$, $(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta}, \{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\}_{\ell\in[q_n]})$ in Game_{q_n} are expressed over bases $\mathbb{D}$ and $\mathbb{D}^*$ as

$$\begin{aligned}
\mathsf{PP} &:= \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \ldots, g_1^{\mathbf{d}_n}\right\},\\
\mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta} &:= \left\{\mathsf{C} := \mathrm{m}\cdot(g_T^\alpha)^z, \quad \mathsf{C}_0 := g_1^{z(x^*_{1,\beta}\mathbf{d}_1+\cdots+x^*_{n,\beta}\mathbf{d}_n)+[\chi_1\mathbf{d}_{n+1}+\cdots+\chi_n\mathbf{d}_{2n}]}\right\},\\
&\left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell} := g_2^{\alpha\mathbf{d}^*_1+r_\ell(v_{1,\ell}\mathbf{d}^*_1+\cdots+v_{n,\ell}\mathbf{d}^*_n)+[\nu_{1,\ell}\mathbf{d}^*_{n+1}+\cdots+\nu_{n,\ell}\mathbf{d}^*_{2n}]}\right\}_{\ell\in[q_n]}.
\end{aligned}$$

Then we can express them over bases $\mathbb{F}$ and $\mathbb{F}^*$ as

$$\begin{aligned}
\mathsf{PP} &:= \left\{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{f}_1}, \ldots, g_1^{\mathbf{f}_n}\right\},\\
\mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta} &:= \left\{\mathsf{C} := \mathrm{m}\cdot(g_T^\alpha)^z, \quad \mathsf{C}_0 := g_1^{(z'_1\mathbf{f}_1+\cdots+z'_n\mathbf{f}_n)+[\chi_1\mathbf{d}_{n+1}+\cdots+\chi_n\mathbf{d}_{2n}]}\right\},\\
&\left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell} := g_2^{\alpha\mathbf{f}^*_1+r_\ell(v_{1,\ell}\mathbf{f}^*_1+\cdots+v_{n,\ell}\mathbf{f}^*_n)+[\nu'_{1,\ell}\mathbf{f}^*_{n+1}+\cdots+\nu'_{n,\ell}\mathbf{f}^*_{2n}]}\right\}_{\ell\in[q_n]},
\end{aligned}$$

where

$$\begin{aligned}
z'_1 &:= zx^*_{1,\beta} - \chi_1\xi_{1,1} - \cdots - \chi_n\xi_{n,1},\\
&\vdots\\
z'_n &:= zx^*_{n,\beta} - \chi_1\xi_{1,n} - \cdots - \chi_n\xi_{n,n},\\
&\left\{\begin{array}{c}\nu'_{1,\ell} := \nu_{1,\ell} + \alpha\xi_{1,1} + r_\ell(v_{1,\ell}\xi_{1,1} + \cdots + v_{n,\ell}\xi_{1,n})\\ \vdots\\ \nu'_{n,\ell} := \nu_{n,\ell} + \alpha\xi_{1,n} + r_\ell(v_{1,\ell}\xi_{n,1} + \cdots + v_{n,\ell}\xi_{n,n})\end{array}\right\}_{\ell\in[q_n]},
\end{aligned}$$

which are all uniformly distributed if $(\chi_1, \ldots, \chi_n)$ defined in Equation 7 is a non-zero vector since $z, \{\xi_{i,j}\}_{i\in[d],j\in[n]}, \{\nu_{1,\ell}, \ldots, \nu_{n,\ell}\}_{\ell\in[q_n]}$ are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients $s(x^*_{1,\beta}, \ldots, x^*_{n,\beta})$ of $\mathbf{d}_1, \ldots, \mathbf{d}_n$ in the C_1 term of the challenge ciphertext is changed to random coefficients $(z'_1, \ldots, z'_n) \in \mathbb{Z}_q^n$ of $\mathbf{f}_1, \ldots, \mathbf{f}_n$, thus the challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$. Moreover, all coefficients $\{(\nu'_{1,\ell}, \ldots, \nu'_{n,\ell})\}_{\ell\in[q_n]}$ of $\mathbf{f}^*_{n+1}, \ldots, \mathbf{f}^*_{2n}$ in the $\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\}_{\ell\in[q_n]}$ are all uniformly distributed since $\{(\nu_{1,\ell}, \ldots, \nu_{n,\ell})\}_{\ell\in[q_n]}$ of $\mathbf{d}^*_{n+1}, \ldots, \mathbf{d}^*_{2n}$ are all independent random values. Thus

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\right\}_{\ell\in[q_n]}\right)$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\mathbf{x}_\mathsf{R}}, \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\right\}_{\ell\in[q_n]}\right)$$

in $\mathsf{Game}_{\mathsf{Final}}$.

In the adversary's view, both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same public parameters. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{q_n} over bases $(\mathbb{D}, \mathbb{D}^*)$ and in $\mathsf{Game}_{\mathsf{Final}}$ over bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, Game_{q_n} and $\mathsf{Game}_{\mathsf{Final}}$ are statistically indistinguishable except with probability $1/q$ (namely, the case $(\chi_1, \ldots, \chi_n) = \mathbf{0}$). □

Lemma 11. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in $\mathsf{Game}_{\mathsf{Final}}$. Hence, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) = 0$. □

In $\mathsf{Game}_{\mathsf{Final}}$, the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$, independent of the two messages and the challenge vectors provided by $\mathcal{A}$. Thus, our IPE scheme is weakly attribute-hiding.

7 Key-Policy Functional Encryption

We now present our KP-FE scheme, the construction and security proof of which are analogues of the DLIN-based KP-FE scheme [34]. Analogously, We define function $\hat{\rho}_1$ by $\hat{\rho}_1(j) := i$ if $\hat{\rho}(j) = (i, \mathbf{v}_j)$ or $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, where $\hat{\rho}$ is given in access structure $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$. As with [34], we only deal with the case that $\hat{\rho}_1$ is injective for $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ with decryption key $\mathsf{SK}_{\mathbb{A}}$ in the proposed KP-FE scheme, see [34] for how to relax the restriction.

Construction. We begin with our KP-FE scheme:

– $\mathsf{Setup}(1^\lambda, \boldsymbol{n} := (d; n_1, \ldots, n_d))$ This algorithm takes in the security parameter λ, a structure $\boldsymbol{n}$ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q. The algorithm samples random dual orthonormal bases, $\{(\mathbb{D}_i, \mathbb{D}_i^*)\}_{i=0,\ldots,d} \leftarrow_{\mathrm{R}} \mathsf{Dual}(\mathbb{Z}_q^3, \mathbb{Z}_q^{2n_1}, \ldots, \mathbb{Z}_q^{2n_d})$. For $i = 0, \ldots, d$ let $\mathbf{d}_{1,i}, \ldots, \mathbf{d}_{2n_i,i}$ denote the elements of $\mathbb{D}_i$ and $\mathbf{d}^*_{1,i}, \ldots, \mathbf{d}^*_{2n_i,i}$ denote the elements of $\mathbb{D}_i^*$, where $2n_0 = 3$. It outputs the public parameters as

$$\mathsf{PP} := \left\{ \mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{ g_1^{\mathbf{d}_{1,i}}, \ldots, g_1^{\mathbf{d}_{n_i,i}} \right\}_{i=1,\ldots,d} \right\} \in G_T \times (G_1^3)^2 \times (G_1^{2n_1})^{n_1} \times \cdots \times (G_1^{2n_d})^{n_d}$$

and the master key

$$\mathsf{MK} := \left\{ g_1^{\mathbf{d}^*_{1,0}}, g_1^{\mathbf{d}^*_{3,0}}, \left\{ g_1^{\mathbf{d}^*_{1,i}}, \ldots, g_1^{\mathbf{d}^*_{n_i,i}} \right\}_{i=1,\ldots,d} \right\} \in (G_2^3)^2 \times (G_2^{2n_1})^{n_1} \times \cdots \times (G_2^{2n_d})^{n_d}.$$

– $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \mathbb{A} := (\hat{\mathbf{A}} \in \mathbb{Z}_q^{\hat{a} \times \hat{b}}, \hat{\rho}))$ This algorithm picks $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\hat{b}}$ and sets $\mathbf{s}^\top := (s_1, \ldots, s_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \mathbf{w}^\top$, $s_0 := \mathbf{1} \cdot \mathbf{w}^\top$. It computes

$$\mathsf{K}_0 := g_2^{-s_0 \mathbf{d}^*_{1,0} + \mathbf{d}^*_{3,0}} \in G_2^3,$$

for $j \in [\hat{a}]$,

- if $\hat{\rho}(j) = (i, \mathbf{v}_j := (v_{1,j}, \ldots, v_{n_i,j}) \in \mathbb{Z}_q^{n_i} \backslash \{\mathbf{0}\})$, it also picks $r_j \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and computes

$$\mathsf{K}_j := g_2^{s_j \mathbf{d}^*_{1,0} + r_j (v_{1,j} \mathbf{d}^*_{1,j} + \cdots + v_{n_i,j} \mathbf{d}^*_{n_i,j})} \in G_2^{2n_i}.$$

- if $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, it computes

$$\mathsf{K}_j := g_2^{s_j (v_{1,j} \mathbf{d}^*_{1,j} + \cdots + v_{n_i,j} \mathbf{d}^*_{n_i,j})} \in G_2^{2n_i}.$$

It returns the secret key $\mathsf{SK}_{\mathbb{A}} := (\mathbb{A}, \mathsf{K}_0, \{\mathsf{K}_j\}_{j \in [\hat{a}]})$.

– $\mathsf{Enc}(\mathsf{PP}, \Gamma := \{(i, \mathbf{x}_i := (x_{1,i}, \ldots, x_{n_i,i})) \in \mathbb{Z}_q^{n_i} \backslash \{\mathbf{0}\} | 1 \le i \le d, x_{1,i} := 1\}, \mathrm{m})$ This algorithm picks $z, z_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$. It computes

$$\begin{aligned} \mathsf{C} &:= \mathrm{m} \cdot g_T^z \in G_T, \quad \mathsf{C}_0 := g_1^{z_0 \mathbf{d}_{1,0} + z \mathbf{d}_{3,0}} \in G_1^3, \\ \mathsf{C}_i &:= g_1^{z_0 (x_{1,i} \mathbf{d}^*_{1,i} + \cdots + x_{n_i,i} \mathbf{d}^*_{n_i,i})} \in G_1^{2n_i} \quad \text{for } (i, \mathbf{x}_i) \in \Gamma. \end{aligned}$$

It returns the ciphertext $\mathsf{CT}_\Gamma := (\Gamma, \mathsf{C}, \mathsf{C}_0, \{\mathsf{C}_i\}_{(i,\mathbf{x}_i) \in \Gamma})$.

– $\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathbb{A}}, \mathsf{CT}_\Gamma)$ This algorithm computes Π and $\{\alpha_j\}_{j \in \Pi}$ such that $s_0 = \sum_{j \in \Pi} \alpha_j s_j$, and

$$\begin{aligned} \Pi \subseteq \{ j \in [\hat{a}] | [\hat{\rho}(j) = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma \wedge \mathbf{x}_i \cdot \mathbf{v}_j = 0] \\ \vee [\hat{\rho}(j) = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma \wedge \mathbf{x}_i \cdot \mathbf{v}_j \neq 0] \}. \end{aligned}$$

It recovers the message as

$$\mathrm{m} := \mathsf{C} / \left(e(\mathsf{C}_0, \mathsf{K}_0) \prod_{j \in \Pi \wedge \hat{\rho}(j) = (i, \mathbf{v}_j)} e(\mathsf{C}_i, \mathsf{K}_j)^{\alpha_j} \prod_{j \in \Pi \wedge \rho(j) = \neg(i, \mathbf{v}_j)} e(\mathsf{C}_i, \mathsf{K}_j)^{\alpha_j / (\mathbf{x}_k \cdot \mathbf{v}_j)} \right) \in G_T$$

Correctness.

$$\begin{aligned} & e(\mathsf{C}_0, \mathsf{K}_0) \prod_{j \in \Pi \wedge \rho(j) = (i, \mathbf{v}_j)} e(\mathsf{C}_i, \mathsf{K}_j)^{\alpha_j} \prod_{j \in \Pi \wedge \rho(j) = \neg(i, \mathbf{v}_j)} e(\mathsf{C}_i, \mathsf{K}_j)^{\alpha_j / (\mathbf{x}_i \cdot \mathbf{v}_j)} \\ &= g_T^{-s_0 z_0 + z} \prod_{j \in \Pi \wedge \rho(j) = (i, \mathbf{v}_j)} g_T^{\alpha_j s_j z_0} \prod_{j \in \Pi \wedge \rho(j) = \neg(i, \mathbf{v}_j)} g_T^{\alpha_j s_j z_0 (\mathbf{x}_i \cdot \mathbf{v}_j) / (\mathbf{x}_i \cdot \mathbf{v}_j)} \\ &= g_T^{z_0 (-s_0 + \sum_{j \in \Pi} \alpha_j s_j) + z} = g_T^z. \end{aligned}$$

Proof of Security. We prove the following theorem by showing a series of lemmas.

Theorem 3. *The KP-FE scheme is fully secure and payload-hiding under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the KP-FE scheme, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_{1^-}, \mathcal{B}_1, \ldots, \mathcal{B}_{q_n^-}, \mathcal{B}_{q_n}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{KP\text{-}FE}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{DDH1}}_{\mathcal{B}_0}(\lambda) + \sum_{\kappa=1}^{q_n} (\mathsf{Adv}^{\mathsf{DDH2}}_{\mathcal{B}_{\kappa^-}}(\lambda) + \mathsf{Adv}^{\mathsf{DDH2}}_{\mathcal{B}_\kappa}(\lambda)) + (2dq_n + 6q_n + d + 3)/q$$

where q_n is the maximum number of $\mathcal{A}$'s key queries.

We still adopt the dual system encryption methodology by Waters [39] to prove the security of our KP-FE scheme. Let $\mathbf{A}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n_i \times n_i}$ ($\mathbf{A}_i$ is invertible with overwhelming probability) for $i \in [d]$:

KeyGenSF A semi-functional key will take on one of two forms. The algorithm first runs normal key generation algorithm to generates

$$\mathsf{SK}_{\mathbb{A}} := (\mathbb{A}, \mathsf{K}_0, \{\mathsf{K}_j\}_{j \in [\hat{a}]}).$$

A semi-functional key of type 1

$$\mathsf{SK}_{\mathbb{A}}^{(\mathsf{SF1})} := (\mathbb{A}, \mathsf{K}_0^{(\mathsf{SF1})}, \{\mathsf{K}_j^{(\mathsf{SF1})}\}_{j \in [\hat{a}]})$$

and a semi-functional key of type 2

$$\mathsf{SK}_{\mathbb{A}}^{(\mathsf{SF2})} := (\mathbb{A}, \mathsf{K}_0^{(\mathsf{SF2})}, \{\mathsf{K}_j^{(\mathsf{SF2})}\}_{j \in [\hat{a}]})$$

are formed as follows. The algorithm also picks $\boldsymbol{\omega} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\hat{b}}$ and sets $\boldsymbol{\theta}^\top := (\theta_1, \ldots, \theta_{\hat{a}})^\top := \mathbf{A} \cdot \boldsymbol{\theta}^\top$. It computes

$$\mathsf{K}_0^{(\mathsf{SF1})} := \mathsf{K}_0 \cdot g_2^{[\theta_0 \mathbf{d}^*_{2,0}]}, \tag{10}$$

$$\mathsf{K}_0^{(\mathsf{SF2})} := \mathsf{K}_0 \cdot g_2^{[\theta_0 \mathbf{d}^*_{2,0}]}, \tag{11}$$

where $\theta_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$. For $j \in [\hat{a}]$,

- if $\hat{\rho}(j) = (i, \mathbf{v}_j := (v_{1,j}, \ldots, v_{n_i,j}) \in \mathbb{Z}_q^{n_i} \backslash \{\mathbf{0}\})$, it also picks $r_j \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and computes

$$\mathsf{K}_j^{(\mathsf{SF1})} := \mathsf{K}_j \cdot g_2^{[\nu_{1,j} \mathbf{d}^*_{n_i+1,i} + \cdots + \nu_{n_i,j} \mathbf{d}^*_{2n_i,i}]}, \tag{12}$$

$$\mathsf{K}_j^{(\mathsf{SF2})} := \mathsf{K}_j, \tag{13}$$

  where $(\nu_{1,j}, \ldots, \nu_{n_i,j}) := \mathbf{A}_i^\top \cdot (\theta_j + \gamma_j v_{1,j}, \gamma_j v_{2,j}, \ldots, \gamma_j v_{n_i,j})^\top$ and $\gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_q$.
- if $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, it computes

$$\mathsf{K}_j^{(\mathsf{SF1})} := \mathsf{K}_j \cdot g_2^{[\nu_{1,j} \mathbf{d}^*_{n_i+1,i} + \cdots + \nu_{n_i,j} \mathbf{d}^*_{2n_i,i}]}, \tag{14}$$

$$\mathsf{K}_j^{(\mathsf{SF2})} := \mathsf{K}_j, \tag{15}$$

  where $(\nu_{1,j}, \ldots, \nu_{n_i,j}) := \theta_j \mathbf{A}_i^\top \cdot \mathbf{v}_j^\top$.

EncryptSF A semi-functional ciphertext will take on one of two forms. The algorithms first run normal key generation algorithm to generate

$$\mathsf{CT}_{\Gamma} := (\Gamma, \mathsf{C}, \mathsf{C}_0, \{\mathsf{C}_i\}_{(i, \mathbf{x}_i) \in \Gamma}).$$

A semi-functional key of type 1

$$\mathsf{CT}_{\Gamma}^{(\mathsf{SF1})} := (\Gamma, \mathsf{C}^{(\mathsf{SF1})}, \mathsf{C}_0^{(\mathsf{SF1})}, \{\mathsf{C}_i\}_{(i, \mathbf{x}_i)^{(\mathsf{SF1})} \in \Gamma})$$

and a semi-functional key of type 2

$$\mathsf{CT}_{\Gamma}^{(\mathsf{SF2})} := (\Gamma, \mathsf{C}^{(\mathsf{SF2})}, \mathsf{C}_0^{(\mathsf{SF2})}, \{\mathsf{C}_i\}_{(i, \mathbf{x}_i)^{(\mathsf{SF2})} \in \Gamma})$$

are formed as follows. It picks $z_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and computes

$$\mathsf{C}^{(\mathsf{SF1})} := \mathsf{C}; \quad \mathsf{C}^{(\mathsf{SF2})} := \mathsf{C}; \tag{16}$$

$$\mathsf{C}_0^{(\mathsf{SF1})} := \mathsf{C}_0 \cdot g_1^{[\zeta_0 \mathbf{d}_{2,0}]}; \quad \mathsf{C}_0^{(\mathsf{SF2})} := \mathsf{C}_0 \cdot g_1^{[\zeta_0 \mathbf{d}_{2,0}]}; \tag{17}$$

$$\left\{ \mathsf{C}_k^{(\mathsf{SF1})} := \mathsf{C}_k \cdot g_1^{[\chi_{1,i} \mathbf{d}^*_{n_i+1,i} + \cdots + \chi_{n_i,i} \mathbf{d}^*_{2n_i,j}]} \right\}_{(i,\mathbf{x}_i) \in \Gamma}; \tag{18}$$

$$\left\{ \mathsf{C}_k^{(\mathsf{SF2})} := \mathsf{C}_k \cdot g_1^{[\chi_{1,i} \mathbf{d}^*_{1,i} + \cdots + \chi_{n_i,i} \mathbf{d}^*_{n_i,j}]} \right\}_{(i,\mathbf{x}_i) \in \Gamma}; \tag{19}$$

where $(\chi_{1,i}, \ldots, \chi_{n_i,i}) := \zeta_0 \mathbf{A}_i^{-1} \cdot \mathbf{x}_i^{\top}$ and $(\chi_{1,i}, \ldots, \chi_{n_i,i}) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n_i}$ in Equations 17 and 18 respectively.

For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q_n key queries $\mathbb{A}_1, \ldots, \mathbb{A}_{q_n}$, our proof of security consists of the following sequence of games between $\mathcal{A}$ and a challenger $\mathcal{B}$.

- $\mathsf{Game}_{\mathsf{Real}}$: is the real security game.
- Game_0: is the same as $\mathsf{Game}_{\mathsf{Real}}$ except that the challenge ciphertext is semi-functional of type 2.
- Game_{κ^-}: for κ from 1 to q_n, Game_κ is the same as Game_0 except that the first $\kappa - 1$ keys are semi-functional of type 2, the κ-th key is semi-functional of type 1, and the remaining keys are normal.
- Game_κ: for κ from 1 to q_n, Game_κ is the same as Game_0 except that the first κ keys are semi-functional of type 2 and the remaining keys are normal.
- $\mathsf{Game}_{\mathsf{Final}}$: is the same as Game_{q_n}, except that the challenge ciphertext is a semi-functional encryption of a random message in G_T. We denote the challenge ciphertext in $\mathsf{Game}_{\mathsf{Final}}$ as $\mathsf{CT}^{(\mathsf{R})}_{\Gamma^*}$.

We prove following lemmas to show the above games are indistinguishable by following an analogous strategy of [34]. Our main arguments are computational indistinguishability (guaranteed by the Subspace assumptions, which are implied by the SXDH assumption) and statistical indistinguishability. The advantage gap between $\mathsf{Game}_{\mathsf{Real}}$ and Game_0 is bounded by the advantage of the Subspace assumption in G_1. Additionally, we require a statistical indistinguishability argument to show that the distribution of the challenge ciphertext remains the same from the adversary's view. For κ from 1 to q_n, the advantage gap between $\mathsf{Game}_{\kappa-1}$ and Game_{κ^-} is bounded by the advantage of Subspace assumption in G_2. Similarly, we require a statistical indistinguishability argument to show that the distribution of the the κ-th semi-functional key remains the same from the adversary's view. For κ from 1 to q_n, the advantage gap between Game_{κ^-} and Game_κ is bounded by the advantage of Subspace assumption in G_2. Finally, we statistically transform Game_{q_n} to $\mathsf{Game}_{\mathsf{Final}}$ in one step, i.e., we show the joint distributions of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF2})}_{\Gamma^*}, \left\{ \mathsf{SK}^{(\mathsf{SF2})}_{\mathbb{A}_\ell} \right\}_{\ell=1,\ldots,q_n} \right) \quad \text{and} \quad \left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\Gamma^*}, \left\{ \mathsf{SK}^{(\mathsf{SF2})}_{\mathbb{A}_\ell} \right\}_{\ell=1,\ldots,q_n} \right)$$

are equivalent for the adversary's view.

We let $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Real}}}_{\mathcal{A}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 12. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Real}}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{B}_0}(\lambda) = \epsilon - (d+2)/q$, with $K_0 = 1$, $\{K_i = n_1\}_{i\in[d]}$ and $N_0 = 3$, $\{N_i = 2n_i\}_{i\in[d]}$.*

Proof. $\mathcal{B}_0$ is given

$$D := \left(\mathbb{G}; g_2^{\mathbf{b}^*_{1,0}}, g_2^{\mathbf{b}^*_{3,0}}, g_1^{\mathbf{b}_{1,0}}, g_1^{\mathbf{b}_{2,0}}, g_1^{\mathbf{b}_{3,0}}, U_{1,0}, \left\{g_2^{\mathbf{b}^*_{1,i}}, \ldots, g_2^{\mathbf{b}^*_{n_i,i}}, g_1^{\mathbf{b}_{1,i}}, \ldots, g_1^{\mathbf{b}_{2n_i,i}}, U_{1,i}, \ldots, U_{1,n_i}\right\}_{i\in[d]}, \mu_2\right)$$

along with $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$. We require that $\mathcal{B}_0$ decides whether $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are distributed as

$$g_1^{\tau_1\mathbf{b}_{1,0}}, \left\{g_1^{\tau_1\mathbf{b}_{1,i}}, \ldots, g_1^{\tau_1\mathbf{b}_{n_i,i}}\right\}_{i\in[d]} \quad \text{or} \quad g_1^{\tau_1\mathbf{b}_{1,0}+\tau_2\mathbf{b}_{2,0}}, \left\{g_1^{\tau_1\mathbf{b}_{1,i}+\tau_2\mathbf{b}_{n_i+1,i}}, \ldots, g_1^{\tau_1\mathbf{b}_{n_i,i}+\tau_2\mathbf{b}_{2n_i,i}}\right\}_{i\in[d]}.$$

$\mathcal{B}_0$ simulates $\mathsf{Game}_{\mathsf{Real}}$ or Game_0 with $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$. To compute the public parameters and master secret key, $\mathcal{B}_0$ implicitly set dual orthonormal bases $\mathbb{D}_0, \mathbb{D}^*_0$ to:

$$\mathbf{d}_{1,0} := \mathbf{b}_{1,0}, \mathbf{d}_{2,0} := \mathbf{b}_{2,0}, \mathbf{d}_{3,0} := \mathbf{b}_{3,0},$$
$$\mathbf{d}^*_{1,0} := \mathbf{b}^*_{1,0}, \mathbf{d}^*_{2,0} := \mathbf{b}^*_{2,0}, \mathbf{d}^*_{3,0} := \mathbf{b}^*_{3,0},$$

for $i \in [d]$, $\mathcal{B}_0$ picks a random invertible matrix $\mathbf{A}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n_i\times n_i}$ and implicitly sets dual orthonormal bases $\mathbb{D}_i, \mathbb{D}^*_i$ to:

$$\mathbf{d}_{1,i} := \mathbf{b}_{1,i}, \ldots, \mathbf{d}_{n_i,i} := \mathbf{b}_{n_i,i}, \quad (\mathbf{d}_{n_i+1,i}, \ldots, \mathbf{d}_{2n_i,i}) := (\mathbf{b}_{n_i+1,i}, \ldots, \mathbf{b}_{2n_i,i})\mathbf{A}_i,$$
$$\mathbf{d}^*_{1,i} := \mathbf{b}^*_{1,i}, \ldots, \mathbf{d}^*_{n_i,i} := \mathbf{b}^*_{n_i,i}, \quad (\mathbf{d}^*_{n_i+1,i}, \ldots, \mathbf{d}^*_{2n_i,i}) := (\mathbf{b}^*_{n_i+1,i}, \ldots, \mathbf{b}^*_{2n_i,i})(\mathbf{A}_i^{-1})^\top.$$

We note that $\{\mathbb{D}_i, \mathbb{D}^*_i\}_{i\in[0,d]}$ are properly distributed, and reveal no information about $\{\mathbf{A}_i\}_{i\in[d]}$. Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}^*_{2,0}}, \{g_2^{\mathbf{d}^*_{n_i+1,i}}, \ldots, g_2^{\mathbf{d}^*_{2n_i,i}}\}_{i\in[d]}$, but these will not be needed for creating normal keys. $\mathcal{B}_0$ computes $g_T := e(g_1, g_2)^{\mathbf{d}_{1,0}\cdot\mathbf{d}^*_{1,0}}$ and then gives $\mathcal{A}$ the public parameters

$$\mathsf{PP} := \left\{\mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{g_1^{\mathbf{d}_{1,i}}, \ldots, g_1^{\mathbf{d}_{n_i,i}}\right\}_{i=1,\ldots,d}\right\},$$

The master key

$$\mathsf{MK} := \left\{g_1^{\mathbf{d}^*_{1,0}}, g_1^{\mathbf{d}^*_{3,0}}, \left\{g_1^{\mathbf{d}^*_{1,i}}, \ldots, g_1^{\mathbf{d}^*_{n_i,i}}\right\}_{i=1,\ldots,d}\right\}$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_0$ the challenge messages $\mathrm{m}_0, \mathrm{m}_1$ and attribute set $\Gamma^* := \{(i, \mathbf{x}_i := (x^*_{1,i}, \ldots, x^*_{n_i,i}))|1 \le i \le d, x_{1,i} := 1\}$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0,1\}$ and encrypts m_β under Γ^* as follows:

$$\mathsf{C} := \mathrm{m}_\beta \cdot g_T^s, \quad \mathsf{C}_0 := T_{1,0} \cdot g_1^{s\mathbf{d}_{3,0}},$$
$$\mathsf{C}_i := T_1^{x^*_{1,i}} \cdots T_{n_i}^{x^*_{n_i,i}} \quad \text{for } (i, \mathbf{x}^*_i) \in \Gamma^*,$$

where $s \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, $\mathcal{B}_0$ has implicitly set $s_0 := \tau_1$. It gives the ciphertext $(\Gamma, \mathsf{C}, \mathsf{C}_0, \{\mathsf{C}_i\}_{(i,\mathbf{x}_i)\in\Gamma})$ to $\mathcal{A}$.

Now, if $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to $g_1^{\tau_1 \mathbf{b}_{1,0}}, \{g_1^{\tau_1 \mathbf{b}_{1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}_{n_i,i}}\}_{i\in[d]}$, then this is a properly distributed normal encryption of m_β. In this case, $\mathcal{B}_0$ has properly simulated $\mathsf{Game}_{\mathsf{Real}}$. If $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to $g_1^{\tau_1 \mathbf{b}_{1,0}+\tau_2 \mathbf{b}_{2,0}}, \{g_1^{\tau_1 \mathbf{b}_{1,i}+\tau_2 \mathbf{b}_{n_i+1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}_{n_i,i}+\tau_2 \mathbf{b}_{2n_i,i}}\}_{i\in[d]}$ instead, then the ciphertext elements $\mathsf{C}_0, \{\mathsf{C}_i\}_{(i,\mathbf{x}_i)\in\Gamma}$ have additional terms of

$$\tau_2 \mathbf{b}_{2,0}, \left\{\tau_2(x^*_{1,i}\mathbf{b}_{n_i+1,i} + \cdots + x^*_{n_i,i}\mathbf{b}_{2n_i,i})\right\}_{(i,\mathbf{x}_i)\in\Gamma}$$

in their exponents. The coefficients here in the basis $\mathbf{b}_{2,0}, \{\mathbf{b}_{n_i+1,i}, \ldots, \mathbf{b}_{2n_i,i}\}_{(i,\mathbf{x}_i)\in\Gamma}$ form the vectors $\tau_2, \{\tau_2(x^*_{1,i}, \ldots, x^*_{n_i,i})\}_{(i,\mathbf{x}_i)\in\Gamma}$. To compute the coefficients in the basis $\mathbf{d}_{2,0}, \{\mathbf{d}_{n_i+1,i}, \ldots, \mathbf{d}_{2n_i,i}\}_{(i,\mathbf{x}_i)\in\Gamma}$, we multiply the matrices $\{\mathbf{A}_i^{-1}\}_{(i,\mathbf{x}_i)\in\Gamma}$ by the transpose of this vectors, obtaining $\tau_2, \{\tau_2 \mathbf{A}_i^{-1}(x^*_{1,i}, \ldots, x^*_{n_i,i})^\top\}_{(i,\mathbf{x}_i)\in\Gamma}$. Since $\{\mathbf{A}_i\}_{i\in[d]}$ are random (everything else given to $\mathcal{A}$ has been distributed independently of $\{\mathbf{A}_i\}_{i\in[d]}$), these coefficients are uniformly random except with probability $(2+d)/q$ (namely, the cases τ_2 defined in Subspace problem is zero, ζ_0 or there exists $i \in [d]$ such that $(\chi_{1,i}, \ldots, \chi_{n_i,i})$ defined in Equations 17 and 19 is zero or the zero vector) from Lemma 1. Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0. This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\mathsf{Game}_{\mathsf{Real}}$ and Game_0 to achieve an advantage ϵ against the Subspace assumption in G_1, namely $\mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{B}_0}(\lambda) = \epsilon - (2+d)/q$. □

Lemma 13. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\kappa-1}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_{\kappa^-}}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa^-}$ such that $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_{\kappa^-}}(\lambda) = \epsilon - 1/q$, with $K_0 = 1$, $\{K_i = n_1\}_{i\in[d]}$ and $N_0 = 3$, $\{N_i = 2n_i\}_{i\in[d]}$.*

Proof. $\mathcal{B}_{\kappa^-}$ is given

$$D := \left(\mathbb{G}; g_1^{\mathbf{b}_{1,0}}, g_1^{\mathbf{b}_{3,0}}, g_1^{\mathbf{b}^*_{1,0}}, g_1^{\mathbf{b}^*_{2,0}}, g_1^{\mathbf{b}^*_{3,0}}, U_{1,0}, \left\{g_1^{\mathbf{b}_{1,i}}, \ldots, g_1^{\mathbf{b}_{n_i,i}}, g_2^{\mathbf{b}^*_{1,i}}, \ldots, g_2^{\mathbf{b}^*_{2n_i,i}}, U_{1,i}, \ldots, U_{n_i,i}\right\}_{i\in[d]}, \mu_2\right)$$

along with $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$. We require that $\mathcal{B}_{\kappa^-}$ decides whether $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are distributed as

$$g_1^{\tau_1 \mathbf{b}^*_{1,0}}, \left\{g_1^{\tau_1 \mathbf{b}^*_{1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}^*_{n_i,i}}\right\}_{i\in[d]} \quad \text{or} \quad g_1^{\tau_1 \mathbf{b}^*_{1,0}+\tau_2 \mathbf{b}^*_{2,0}}, \left\{g_1^{\tau_1 \mathbf{b}^*_{1,i}+\tau_2 \mathbf{b}^*_{n_i+1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}^*_{n_i,i}+\tau_2 \mathbf{b}^*_{2n_i,i}}\right\}_{i\in[d]}.$$

$\mathcal{B}_{\kappa^-}$ simulates Game_{κ^-} or $\mathsf{Game}_{\kappa-1}$ with $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$. To compute the public parameters and master secret key, $\mathcal{B}_{\kappa^-}$ implicitly set dual orthonormal bases $\mathbb{D}_0, \mathbb{D}^*_0$ to:

$$\mathbf{d}_{1,0} := \mathbf{b}_{1,0}, \mathbf{d}_{2,0} := \mathbf{b}_{2,0}, \mathbf{d}_{3,0} := \mathbf{b}_{3,0},$$
$$\mathbf{d}^*_{1,0} := \mathbf{b}^*_{1,0}, \mathbf{d}^*_{2,0} := \mathbf{b}^*_{2,0}, \mathbf{d}^*_{3,0} := \mathbf{b}^*_{3,0},$$

for $i \in [d]$, $\mathcal{B}_{\kappa^-}$ picks a random invertible matrix $\mathbf{A}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n_i \times n_i}$ and implicitly set dual orthonormal bases $\mathbb{D}_i, \mathbb{D}^*_i$ to:

$$\mathbf{d}_{1,i} := \mathbf{b}_{1,i}, \ldots, \mathbf{d}_{n_i,i} := \mathbf{b}_{n_i,i}, \quad (\mathbf{d}_{n_i+1,i}, \ldots, \mathbf{d}_{2n_i,i}) := (\mathbf{b}_{n_i+1,i}, \ldots, \mathbf{b}_{2n_i,i})\mathbf{A}_i,$$
$$\mathbf{d}^*_{1,i} := \mathbf{b}^*_{1,i}, \ldots, \mathbf{d}^*_{n_i,i} := \mathbf{b}^*_{n_i,i}, \quad (\mathbf{d}^*_{n_i+1,i}, \ldots, \mathbf{d}^*_{2n_i,i}) := (\mathbf{b}^*_{n_i+1,i}, \ldots, \mathbf{b}^*_{2n_i,i})(\mathbf{A}_i^{-1})^\top.$$

We note that $\{\mathbb{D}_i, \mathbb{D}_i^*\}_{i\in[0,d]}$ are properly distributed, and reveal no information about $\{\mathbf{A}_i\}_{i\in[d]}$. $\mathcal{B}_{\kappa^-}$ computes $g_T := e(g_1, g_2)^{\mathbf{d}_{1,0}\cdot\mathbf{d}^*_{1,0}}$ and then gives $\mathcal{A}$ the public parameters

$$\mathsf{PP} := \left\{\mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{g_1^{\mathbf{d}_{1,i}}, \ldots, g_1^{\mathbf{d}_{n_i,i}}\right\}_{i=1,\ldots,d}\right\},$$

The master key

$$\mathsf{MK} := \left\{g_1^{\mathbf{d}^*_{1,0}}, g_1^{\mathbf{d}^*_{3,0}}, \left\{g_1^{\mathbf{d}^*_{1,i}}, \ldots, g_1^{\mathbf{d}^*_{n_i,i}}\right\}_{i=1,\ldots,d}\right\}$$

is known to $\mathcal{B}_{\kappa^-}$, which allows $\mathcal{B}_{\kappa^-}$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa^-}$ also knows $g_2^{\mathbf{d}^*_{2,0}}$, it can easily produce semi-functional keys of type 2.

To answer the first $\kappa-1$ key queries that $\mathcal{A}$ makes, $\mathcal{B}_{\kappa^-}$ runs the semi-functional key generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ-th key query for $\mathbb{A}_\kappa := (\hat{\mathbf{A}}, \hat{\rho})$, $\mathcal{B}_{\kappa^-}$ picks $\mathbf{w}', \boldsymbol{\omega} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\hat{b}}$ and sets $\mathbf{s}'^{\top} := (s'_1, \ldots, s'_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \mathbf{w}'^{\top}$,$\boldsymbol{\theta}'^{\top} := (\theta'_1, \ldots, \theta'_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \boldsymbol{\omega}'^{\top}$, $s'_0 := \mathbf{1} \cdot \mathbf{w}'^{\top}$, $\theta'_0 := \mathbf{1} \cdot \boldsymbol{\omega}'^{\top}$. It computes

$$\mathsf{K}_0 := T_{1,0}^{-\theta'_0} \cdot g_2^{-s'_0\mathbf{d}^*_{1,0}+\mathbf{d}^*_{3,0}},$$

for $j \in [\hat{a}]$,

- if $\hat{\rho}(j) = (i, \mathbf{v}_j := (v_{1,j}, \ldots, v_{n_i,j}) \in \mathbb{Z}_q^{n_i}\backslash\{\mathbf{0}\})$, it also picks $r'_j, \gamma' \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and computes

$$\mathsf{K}_j := T_{1,i}^{\theta'_j} \cdot (T_{1,i}^{v_{1,j}} \cdots T_{n_i,i}^{v_{n_i,j}})^{\gamma'} \cdot g_2^{s'_j\mathbf{d}^*_1 + r'_j(v_{1,j}\mathbf{d}^*_{1,j}+\cdots+v_{n_i,j}\mathbf{d}^*_{n_i,j})}.$$

- if $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, it computes

$$\mathsf{K}_j := (T_{1,i}^{v_{1,j}} \cdots T_{n_i,i}^{v_{n_i,j}})^{\theta'_j} \cdot g_2^{s'_j(v_{1,j}\mathbf{d}^*_{1,j}+\cdots+v_{n_i,j}\mathbf{d}^*_{n_i,j})}.$$

This implicitly sets

$$\mathbf{s}^\top := (s_1, \ldots, s_{\hat{a}})^\top := \mathbf{s}' + \tau_1\boldsymbol{\theta}',\ s_0 := s'_0 + \tau_1\theta'_0,\ \{r_j := r'_j + \tau_1\gamma'_j,\ \gamma_j := \tau_2\gamma'_j\}_{j\in[\hat{a}]}.$$

We note that if $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to

$$g_1^{\tau_1\mathbf{b}^*_{1,0}+\tau_2\mathbf{b}^*_{2,0}}, \left\{g_1^{\tau_1\mathbf{b}^*_{1,i}+\tau_2\mathbf{b}^*_{n_i+1,i}}, \ldots, g_1^{\tau_1\mathbf{b}^*_{n_i,i}+\tau_2\mathbf{b}^*_{2n_i,i}}\right\}_{i\in[d]},$$

whose exponent vectors include

$$-\tau_2\theta'_0\mathbf{b}^*_{2,0}, \left\{\begin{array}{ll}\tau_2(\theta'_j\mathbf{b}^*_{n_i,i} + \gamma'_j(v_{1,j}\mathbf{b}^*_{n_i+1,i} + \cdots + v_{n_i,i}\mathbf{b}^*_{2n_i})), & \text{if } \hat{\rho}(j) = (i, \mathbf{v}_j)\\ \tau_2\theta'_j(v_{1,j}\mathbf{b}^*_{n_i+1,i} + \cdots + v_{n_i,i}\mathbf{b}^*_{2n_i}), & \text{if } \hat{\rho}(j) = \neg(i, \mathbf{v}_j)\end{array}\right\}_{j\in[\hat{a}]}. \tag{20}$$

We take the coefficients of vectors in Equation 20 in terms of the bases $\mathbf{b}^*_{2,0}, \{\mathbf{b}^*_{n_i+1,i}, \ldots, \mathbf{b}^*_{2n_i,i}\}_{i\in[d]}$ and translate them into coefficients in terms of the bases $\mathbf{d}^*_{2,0}, \{\mathbf{d}^*_{n_i+1,i}, \ldots, \mathbf{d}^*_{2n_i,i}\}_{i\in[d]}$. Using the change of basis matrix $\{\mathbf{A}_i\})_{i\in[d]}$, we obtain the new coefficients (in vector form) as:

$$-\tau_2\theta'_0, \left\{\begin{array}{ll}\mathbf{A}_i^\top(\tau_2\theta'_j + \tau_2\gamma'_j v_{1,j}, \ldots, \tau_2\gamma'_j v_{n_i,i})^\top, & \text{if } \hat{\rho}(j) = (i, \mathbf{v}_j)\\ \tau_2\theta'_j\mathbf{A}_i^\top(v_{1,j}, \ldots, v_{n_i,i})^\top, & \text{if } \hat{\rho}(j) = \neg(i, \mathbf{v}_j)\end{array}\right\}_{j\in[\hat{a}]}. \tag{21}$$

We will argue that this is a semi-functional key of type 1. To respond to the remaining key queries, $\mathcal{B}_{\kappa^-}$ simply runs the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_{\kappa^-}$ the challenge messages $\mathrm{m}_0, \mathrm{m}_1$ and attribute set $\Gamma^* := \{(i, \mathbf{x}_i := (x^*_{1,i}, \ldots, x^*_{n_i,i}))|1 \leq i \leq d, x_{1,i} := 1\}$. $\mathcal{B}_{\kappa^-}$ chooses a random bit $\beta \in \{0,1\}$ and encrypts m_β under Γ^* as follows:

$$\mathsf{C} := \mathrm{m}_\beta \cdot g_T^z, \quad \mathsf{C}_0 := U_{1,0} \cdot g_1^{z\mathbf{d}_{3,0}},$$
$$\left\{\mathsf{C}_i := (U_{1,i})^{x^*_{1,i}\mathbf{d}_{1,i}} \cdots (U_{n_i,i})^{x^*_{n_i,i}\mathbf{d}_{n_i,i}}\right\}_{(i,\mathbf{x}^*_i)\in\Gamma^*},$$

where $z \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, $\mathcal{B}_{\kappa^-}$ has implicitly set $z_0 := \mu_1$. It gives the ciphertext $(\Gamma, \mathsf{C}, \mathsf{C}_0, \{\mathsf{C}_i\}_{(i,\mathbf{x}_i)\in\Gamma})$ to $\mathcal{A}$. The "semi-functional part" of the exponent vector here is:

$$\mu_2\mathbf{b}_{2,0}, \ \left\{\mu_2(x_{1,i}\mathbf{b}^*_{n_i+1,i} + \cdots + x_{n_i,i}\mathbf{b}^*_{2n_i})\right\}_{(i,\mathbf{x}^*_i)\in\Gamma^*}. \tag{22}$$

We take the coefficients of vectors in Equation 22 in terms of the bases $\mathbf{b}_{2,0}, \{\mathbf{b}_{n_i+1,i}, \ldots, \mathbf{b}_{2n_i,i}\}_{i\in[d]}$ and translate them into coefficients in terms of the bases $\mathbf{d}_{2,0}, \{\mathbf{d}_{n_i+1,i}, \ldots, \mathbf{d}_{2n_i,i}\}_{i\in[d]}$. Using the change of basis matrix $\{\mathbf{A}_i\})_{i\in[d]}$, we obtain the new coefficients (in vector form) as:

$$\mu_2, \ \left\{\mu_2\mathbf{A}_i^{-1}(x_{1,i}, \ldots, x_{n_i,i})\right\}_{(i,\mathbf{x}^*_i)\in\Gamma^*}. \tag{23}$$

If $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to

$$g_1^{\tau_1\mathbf{b}^*_{1,0}}, \left\{g_1^{\tau_1\mathbf{b}^*_{1,i}}, \ldots, g_1^{\tau_1\mathbf{b}^*_{n_i,i}}\right\}_{i\in[d]},$$

then the κ-th responding key query is a properly distributed normal key while the vectors in Equation 23 is a semi-functional ciphertext of type 2 except with probability $(2+d)/q$ (namely, the cases μ_2 defined in Subspace problem is zero, ζ_0 or there exists $i \in [d]$ such that $(\chi_{1,i}, \ldots, \chi_{n_i,i})$ defined in Equations 17 and 19 is zero or the zero vector) from Lemma 1. Thus, $\mathcal{B}_{\kappa^-}$ has properly simulated $\mathsf{Game}_{\kappa-1}$.

If $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to

$$g_1^{\tau_1\mathbf{b}^*_{1,0}+\tau_2\mathbf{b}^*_{2,0}}, \left\{g_1^{\tau_1\mathbf{b}^*_{1,i}+\tau_2\mathbf{b}^*_{n_i+1,i}}, \ldots, g_1^{\tau_1\mathbf{b}^*_{n_i,i}+\tau_2\mathbf{b}^*_{2n_i,i}}\right\}_{i\in[d]},$$

we now argue that $\mathcal{B}_{\kappa^-}$ has properly simulated Game_{κ^-}. Observe that the vectors in Equations 21 and 23 are distributed as semi-functional key and ciphertext of type 1 except that $\tau_2\theta'_0 := \tau_2 \cdot \mathbf{1} \cdot \boldsymbol{\omega}'$ is related to $\tau_2(\theta'_1, \ldots, \theta'_{\hat{a}}) := \tau_2\hat{\mathbf{A}} \cdot \boldsymbol{\omega}'$ instead of a random value from $\mathbb{Z}_q$. We claim that since $\mathbb{A}_\kappa$ does not accept Γ^*, in $\mathcal{A}$'s view the vectors in Equations 20 and 22 are well distributed. Note that for any $j \in [\hat{a}]$, $(\mathbf{A}_i^\top, \mathbf{A}_i^{-1})$ with $i := \hat{\rho}_1(j)$ is independent from the other variables, since $\hat{\rho}_1$ is injective:

1. $[\hat{\rho} = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma^* \wedge \mathbf{x}_i \cdot \mathbf{v}_j = 0]$.
   Then, from Lemma 1, the joint distribution of

   $$\left(\mu_2\mathbf{A}_i^{-1}(x_{1,i}, \ldots, x_{n_i,i}), \mathbf{A}_i^\top(\tau_2\theta'_j + \tau_2\gamma'_j v_{1,j}, \ldots, \tau_2\gamma'_j v_{n_i,i})^\top\right)$$

   is uniformly and independently distributed on $C_{\tau_2\theta'_i\mu_2} := \{(\mathbf{z}, \mathbf{w})|\mathbf{z} \cdot \mathbf{w} = \tau_2\theta'_i\mu_2\}$.

2. $[\hat{\rho} = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma^* \wedge \mathbf{x}_i \cdot \mathbf{v}_j \neq 0]$.
   Then, from Lemma 1, the joint distribution of

$$\left(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}, \ldots, x_{n_i,i}), \tau_2 \theta'_j \mathbf{A}_i^{\top}(v_{1,j}, \ldots, v_{n_i,i})^{\top}\right)$$

   is uniformly and independently distributed on $C_{\tau_2 \theta'_i \mu_2} := \{(\mathbf{z}, \mathbf{w}) | \mathbf{z} \cdot \mathbf{w} = \tau_2 \theta'_i \mu_2\}$.
3. $[\hat{\rho} = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma^* \wedge \mathbf{x}_i \cdot \mathbf{v}_j \neq 0]$.
   Then, from Lemma 1, the joint distribution of

$$\left(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}, \ldots, x_{n_i,i}), \mathbf{A}_i^{\top}(\tau_2 \theta'_j + \tau_2 \gamma'_j v_{1,j}, \ldots, \tau_2 \gamma'_j v_{n_i,i})^{\top}\right)$$

   is uniformly and independently distributed on $C_{\mu_2 \tau_2 \theta'_i + \tau_2 \theta'_j \mu_2 \mathbf{x}_i \cdot \mathbf{v}_j}$. Since γ'_j is uniformly and independently distributed on $\mathbb{Z}_q$, the above joint distribution is uniformly and independently distributed over $\mathbb{Z}_q^{2n_i}$.
4. $[\hat{\rho} = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma^* \wedge \mathbf{x}_i \cdot \mathbf{v}_j = 0]$.
   Then, from Lemma 1, the joint distribution of

$$\left(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}, \ldots, x_{n_i,i}), \tau_2 \theta'_j \mathbf{A}_i^{\top}(v_{1,j}, \ldots, v_{n_i,i})^{\top}\right)$$

   is uniformly and independently distributed on C_0.
5. $[\hat{\rho} = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \notin \Gamma^*]$ or $[\hat{\rho} = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \notin \Gamma^*]$.
   Then, the distribution of

$$\left(\mathbf{A}_i^{\top}(\tau_2 \theta'_j + \tau_2 \gamma'_j v_{1,j}, \ldots, \tau_2 \gamma'_j v_{n_i,i})^{\top} \text{ or } \tau_2 \theta'_j \mathbf{A}_i^{\top}(v_{1,j}, \ldots, v_{n_i,i})^{\top}\right)$$

   is uniformly and independently distributed on $\mathbb{Z}_q^{n_i}$.

We then observe the joint distribution (or relation) of 21 and 23. Those in cases 3-5 are obviously independent from $\tau_2 \theta'_0$. Due to the restriction of adversary $\mathcal{A}$'s key queries, $\mathbb{A}_\kappa$ does not accept Γ^*. Therefore, $\tau_2 \theta'_0 := \tau_2 \cdot \mathbf{1} \cdot \boldsymbol{\omega'}$ is independent from the joint distribution of $\tau_2(\theta'_1, \ldots, \theta'_{\hat{a}}) := \tau_2 \hat{\mathbf{A}} \cdot \boldsymbol{\omega'}$ (over the random selection of $\boldsymbol{\omega'}$). Thus, $\tau_2 \theta'_0$ is uniformly and independently distributed from the other variables in the joint distribution of $\mathcal{B}_{\kappa^-}$'s simulation except with probability $1/q$ (namely, the case τ_2 defined in Subspace problem is zero) from Lemma 1.

In summary, $\mathcal{B}_{\kappa^-}$ has properly simulated either $\mathsf{Game}_{\kappa-1}$ or Game_{κ^-} for $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i \in [d]}$. It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - (d+3)/q$ against the Subspace assumption in G_2, namely $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_{\kappa^-}}(\lambda) = \epsilon - (d+3)/q$. □

Lemma 14. *Suppose that there exists an adversary $\mathcal{A}$ where $|\mathsf{Adv}^{\mathsf{Game}_{\kappa^-}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_{\kappa}}_{\mathcal{A}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_\kappa$ such that $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_\kappa}(\lambda) = \epsilon - (d+3)/q$, with $K_0 = 1$, $\{K_i = n_1\}_{i \in [d]}$ and $N_0 = 3$, $\{N_i = 2n_i\}_{i \in [d]}$.*

Proof. $\mathcal{B}_\kappa$ is given

$$D := \left(\mathbb{G}; g_1^{\mathbf{b}_{1,0}}, g_1^{\mathbf{b}_{3,0}}, g_1^{\mathbf{b}^*_{1,0}}, g_1^{\mathbf{b}^*_{2,0}}, g_1^{\mathbf{b}^*_{3,0}}, U_{1,0}, \{g_1^{\mathbf{b}_{1,i}}, \ldots, g_1^{\mathbf{b}_{n_i,i}}, g_2^{\mathbf{b}^*_{1,i}}, \ldots, g_2^{\mathbf{b}^*_{2n_i,i}}, U_{1,i}, \ldots, U_{n_i,i}\}_{i \in [d]}, \mu_2\right)$$

along with $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$. We require that $\mathcal{B}_\kappa$ decides whether $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are distributed as

$$g_1^{\tau_1 \mathbf{b}^*_{1,0}}, \left\{ g_1^{\tau_1 \mathbf{b}^*_{1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}^*_{n_i,i}} \right\}_{i\in[d]} \quad \text{or} \quad g_1^{\tau_1 \mathbf{b}^*_{1,0} + \tau_2 \mathbf{b}^*_{2,0}}, \left\{ g_1^{\tau_1 \mathbf{b}^*_{1,i} + \tau_2 \mathbf{b}^*_{n_i+1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}^*_{n_i,i} + \tau_2 \mathbf{b}^*_{2n_i,i}} \right\}_{i\in[d]}.$$

$\mathcal{B}_\kappa$ acts in the same way as $\mathcal{B}_{\kappa^-}$ in the proof of Lemma 13 except that K_0 is responded as:

$$\mathsf{K}_0 := T_{1,0}^{-\theta'_0} \cdot g_2^{-s'_0 \mathbf{d}^*_{1,0} + \mathbf{d}^*_{3,0}} \cdot g_2^{\theta''_0 \mathbf{d}^*_{2,0}},$$

where $\theta''_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$.

Now, if $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}_{1,0}}, \left\{ g_1^{\tau_1 \mathbf{b}_{1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}_{n_i,i}} \right\}_{i\in[d]},$$

then the κ-th responding key query is a properly distributed normal key while the challenge ciphertext is a semi-functional ciphertext of type 2 except with probability $(2+d)/q$ (namely, the cases μ_2 defined in Subspace problem is zero, ζ_0 or there exists $i \in [d]$ such that $(\chi_{1,i}, \ldots, \chi_{n_i,i})$ defined in Equations 17 and 19 is zero or the zero vector) from Lemma 1. In this case, $\mathcal{B}_\kappa$ has properly simulated Game_{κ^-}.

If $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}^*_{1,0}}, \left\{ g_1^{\tau_1 \mathbf{b}^*_{1,i}}, \ldots, g_1^{\tau_1 \mathbf{b}^*_{n_i,i}} \right\}_{i\in[d]},$$

then the κ-th responding key query is a properly distributed semi-functional key of type 1 while the challenge ciphertext is a semi-functional ciphertext of type 1 except with probability $1/q$ (namely, the case τ_2 defined in Subspace problem is zero) from Lemma 1. Thus, $\mathcal{B}_{\kappa^-}$ has properly simulated $\mathsf{Game}_{\kappa-1}$.

In summary, $\mathcal{B}_\kappa$ has properly simulated either Game_{κ^-} or Game_κ for $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \ldots, T_{n_i,i}\}_{i\in[d]}$. It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - (d+3)/q$ against the Subspace assumption in G_2, namely $\mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_\kappa}(\lambda) = \epsilon - (d+3)/q$. □

Lemma 15. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Game}_{\mathsf{Final}}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_{q_n}}_{\mathcal{A}}(\lambda) + 1/q$.*

Proof. To prove this lemma, we show the joint distributions of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF2})}_{\Gamma^*}, \left\{ \mathsf{SK}^{(\mathsf{SF2})}_{\mathbb{A}_\ell} \right\}_{\ell=1,\ldots,q_n} \right)$$

in Game_{q_n} and that of

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\Gamma^*}, \left\{ \mathsf{SK}^{(\mathsf{SF2})}_{\mathbb{A}_\ell} \right\}_{\ell=1,\ldots,q_n} \right)$$

in $\mathsf{Game}_{\mathsf{Final}}$ are equivalent for the adversary's view, where $\mathsf{CT}^{(\mathsf{R})}_{\Gamma^*}$ is a semi-functional encryption of a random message in G_T.

By definition, we only need to consider elements based on $\mathbb{D}_0 := (\mathbf{d}_{1,0}, \mathbf{d}_{2,0}, \mathbf{d}_{3,0})$ and $\mathbb{D}_0^* := (\mathbf{d}_{1,0}^*, \mathbf{d}_{2,0}^*, \mathbf{d}_{3,0}^*)$. For this purpose, we pick $\xi \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and define new dual orthonormal bases $\mathbb{F}_0 := (\mathbf{f}_{1,0}, \mathbf{f}_{2,0}, \mathbf{f}_{3,0})$, and $\mathbb{F}_0^* := (\mathbf{f}_{1,0}^*, \mathbf{f}_{2,0}^*, \mathbf{f}_{3,0}^*)$ as follows:

$$\begin{pmatrix} \mathbf{f}_{1,0} \\ \mathbf{f}_{2,0} \\ \mathbf{f}_{3,0} \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \xi \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{1,0} \\ \mathbf{d}_{2,0} \\ \mathbf{d}_{3,0} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{f}_{1,0}^* \\ \mathbf{f}_{2,0}^* \\ \mathbf{f}_{3,0}^* \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\xi & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{1,0}^* \\ \mathbf{d}_{2,0}^* \\ \mathbf{d}_{3,0}^* \end{pmatrix}.$$

It is easy to verify that $\mathbb{F}_0$ and $\mathbb{F}_0^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}_0$ and $\mathbb{D}_0^*$.

Then the public parameters, challenge ciphertext for Γ^*, and queried secret keys for $\{\mathbb{A}_\ell\}_{\ell=1,\ldots,q_n}$, $(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF})}_{\mathbf{x}^*_\beta}, \{\mathsf{SK}^{(\mathsf{SF})}_{\mathbf{v}_\ell}\}_{\ell=1,\ldots,q_n})$ in Game_{q_n} are expressed over bases $\mathbb{D}_0$ and $\mathbb{D}_0^*$ as

$$\begin{aligned} \mathsf{PP} &:= \left\{\mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \ldots\right\} \\ \mathsf{CT}^{(\mathsf{SF})}_{\Gamma^*} &:= \left\{\mathsf{C} := \mathrm{m} \cdot g_T^z, \quad \mathsf{C}_0 := g_1^{z_0\mathbf{d}_{1,0} + \zeta_0\mathbf{d}_{2,0} + z\mathbf{d}_{3,0}}, \ldots\right\}, \\ \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbb{A}_\ell} := \left\{\mathbb{A}_\ell, \mathsf{K}_{0,\ell} := g_2^{-s_{0,\ell}\mathbf{d}_{1,0}^* + \theta_{0,\ell}\mathbf{d}_{2,0}^* + \mathbf{d}_{3,0}^*}, \ldots\right\}\right\}_{\ell=1,\ldots,q_n}. \end{aligned}$$

Then we can express them over bases $\mathbb{F}_0$ and $\mathbb{F}_0^*$ as

$$\begin{aligned} \mathsf{PP} &:= \left\{\mathbb{G}; g_T, g_1^{\mathbf{f}_{1,0}}, g_1^{\mathbf{f}_{3,0}}, \ldots\right\} \\ \mathsf{CT}^{(\mathsf{SF})}_{\Gamma^*} &:= \left\{\mathsf{C} := \mathrm{m} \cdot g_T^z, \quad \mathsf{C}_0 := g_1^{z_0\mathbf{f}_{1,0} + \zeta_0\mathbf{f}_{2,0} + z'\mathbf{f}_{3,0}}, \ldots\right\}, \\ \left\{\mathsf{SK}^{(\mathsf{SF})}_{\mathbb{A}_\ell} := \left\{\mathbb{A}_\ell, \mathsf{K}_{0,\ell} := g_2^{-s_{0,\ell}\mathbf{f}_{1,0}^* + \theta'_{0,\ell}\mathbf{f}_{2,0}^* + \mathbf{f}_{3,0}^*}, \ldots\right\}\right\}_{\ell=1,\ldots,q_n} \end{aligned}$$

where

$$\begin{aligned} z' &:= z - \zeta_0\xi, \\ \left\{\theta'_{0,\ell} := \theta_{0,\ell} + \xi\right\}_{\ell\in[q_n]}, \end{aligned}$$

which are all uniformly distributed if $\zeta_0 \neq 0$ since $\xi, \{\theta_{0,\ell}\}_{\ell\in[q_n]}$ are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients (z_0, ζ_0, z) of $\mathbf{d}_{0,1}, \mathbf{d}_{0,2}, \mathbf{d}_{0,3}$ in the C_0 term of the challenge ciphertext is changed to coefficients $(z_0, \zeta_0, z') \in \mathbb{Z}_q^3$ of $\mathbf{f}_{1,0}, \mathbf{f}_{2,0}, \mathbf{f}_{3,0}$. The challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T, since z in the C is hidden. Moreover, all other coefficients are all well distributed. Thus

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{SF2})}_{\Gamma^*}, \left\{\mathsf{SK}^{(\mathsf{SF2})}_{\mathbb{A}_\ell}\right\}_{\ell=1,\ldots,q_n}\right)$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$\left(\mathsf{PP}, \mathsf{CT}^{(\mathsf{R})}_{\Gamma^*}, \left\{\mathsf{SK}^{(\mathsf{SF2})}_{\mathbb{A}_\ell}\right\}_{\ell=1,\ldots,q_n}\right)$$

in $\mathsf{Game}_{\mathsf{Final}}$.

In the adversary's view, both $(\mathbb{D}_0, \mathbb{D}_0^*)$ and $(\mathbb{F}_0, \mathbb{F}_0^*)$ are consistent with the same public parameters. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{q_n} over bases $(\mathbb{D}_0, \mathbb{D}_0^*)$ and in $\mathsf{Game}_{\mathsf{Final}}$ over bases $(\mathbb{F}_0, \mathbb{F}_0^*)$. Thus, Game_{q_n} and $\mathsf{Game}_{\mathsf{Final}}$ are statistically indistinguishable except with probability $1/q$ (namely, the case $\zeta_0 = 0$). □

Lemma 16. *For any adversary* $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_{\mathsf{Final}}}(\lambda) = 0$.

Proof. The value of β is independent from the adversary's view in $\mathsf{Game}_{\mathsf{Final}}$. Hence, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_{\mathsf{Final}}}(\lambda) = 0$. □

8 Conclusion

In this paper, we presented Subspace assumptions derived from the SXDH assumption. We also instantiate our framework to (anonymous) IBE, IPE, and KP-FE schemes. By shifting from the DLIN assumption to the simpler SXDH assumption, our schemes that are syntactically simpler and achieve shorter parameters.

Acknowledgments. We thank David Freeman, Kenny Paterson and the anonymous referees for helpful comments on an earlier draft of this paper.

References

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

[2] G. Ateniese, J. Kirsch, and M. Blanton. Secret handshakes with dynamic and fuzzy matching. In *NDSS*, 2007.

[3] L. Ballard, M. Green, B. de Medeiros, and F. Monrose. Correlation-resistant storage via keyword-searchable encryption. IACR Cryptology ePrint Archive, Report 2005/417, 2005.

[4] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for key management—part 1: General (revised). *NIST Special Pub*, 800-57, 2007.

[5] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, pages 319–331, 2005.

[6] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. Ph.D., Technion - Israel Institute of Technology, 1996.

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

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

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

[10] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[11] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[12] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[13] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.

[14] Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS*, pages 501–510, 2010.

[15] F. Brezing and A. Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37(1):133–141, 2005.

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

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

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

[19] L. Ducas. Anonymity from asymmetry: New constructions for anonymous HIBE. In *CT-RSA*, pages 148–164, 2010.

[20] D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2):224–280, 2010.

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

[22] S. D. Galbraith and V. Rotger. Easy decision Diffie-Hellman groups. IACR Cryptology ePrint Archive, Report 2004/070, 2004.

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

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

[25] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[26] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

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

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

[29] 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.

[30] 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.

[31] A. Miyaji, M. Nakabayashi, and S. Takano. Characterization of elliptic curve traces under fr-reduction. In *ICISC*, pages 90–108, 2000.

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

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

[34] 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.
[35] 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.
[36] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
[37] E. R. Verheul. Evidence that XTR is more secure than Supersingular Elliptic Curve cryptosystems. *Journal of Cryptology*, 17(4):277–296, 2004.
[38] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
[39] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

A Hierarchical Identity-Based Encryption

We present a HIBE scheme, which is essentially a special case of our KP-FE.

Construction. We begin with our HIBE scheme:

– $\mathsf{Setup}(1^\lambda, d)$ This algorithm takes in the security parameter λ, a depth parameter d and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q. The algorithm samples random dual orthonormal bases, $\{(\mathbb{D}_i, \mathbb{D}_i^*)\}_{i=0,\ldots,d} \leftarrow_{\mathrm{R}} \mathsf{Dual}(\mathbb{Z}_q^3, \mathbb{Z}_q^4, \ldots, \mathbb{Z}_q^4)$. It outputs the public parameters as

$$\mathsf{PP} := \left\{ \mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{ g_1^{\mathbf{d}_{1,i}}, g_1^{\mathbf{d}_{2,i}} \right\}_{i=1,\ldots,d}, g_1^{\mathbf{d}^*_{1,0}}, \left\{ g_1^{\mathbf{d}^*_{1,i}}, g_1^{\mathbf{d}^*_{2,i}} \right\}_{i=1,\ldots,d} \right\} \in G_T \times (G_1^3)^2 \times (G_1^4)^{2d} \times G_2^3 \times (G_2^4)^{2d}$$

and the master key

$$\mathsf{MK} := g_1^{\mathbf{d}^*_{3,0}} \in G_2^3.$$

– $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, (\mathrm{id}_1, \ldots, \mathrm{id}_\ell))$ This algorithm picks $r_1, \ldots, r_\ell, s_1, \ldots, s_\ell \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and sets $s_0 := s_1 + \cdots + s_\ell$. The secret key is computed as

$$\mathsf{SK}_{(\mathrm{id}_1,\ldots,\mathrm{id}_\ell)} := \left\{ \mathsf{K}_0 := g_2^{-s_0 \mathbf{d}^*_{1,0} + \mathbf{d}^*_{3,0}}, \left\{ \mathsf{K}_i := g_2^{s_i \mathbf{d}^*_{1,i} + r_i(\mathrm{id}_i \mathbf{d}^*_{1,i} - \mathbf{d}^*_{2,i})} \right\}_{i=1,\ldots,\ell} \right\} \in G_2^3 \times (G_2^4)^\ell.$$

– $\mathsf{Enc}(\mathsf{PP}, (\mathrm{id}_1, \ldots, \mathrm{id}_\ell), \mathrm{m})$ This algorithm picks $z, z_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\mathsf{CT}_{(\mathrm{id}_1,\ldots,\mathrm{id}_\ell)} := \left\{ \mathsf{C} := \mathrm{m} \cdot g_T^z, \mathsf{C}_0 := g_1^{z_0 \mathbf{d}_{1,0} + z \mathbf{d}_{3,0}}, \left\{ \mathsf{C}_i := g_1^{z_0(\mathbf{d}_{1,i} + \mathrm{id}_i \mathbf{d}_{2,i})} \right\}_{i=1,\ldots,\ell} \right\} \in G_T \times G_1^3 \times (G_1^4)^\ell.$$

– $\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{(\mathrm{id}_1,\ldots,\mathrm{id}_\ell)}, \mathsf{CT}_{(\mathrm{id}_1,\ldots,\mathrm{id}_\ell)})$ This algorithm computes the message as

$$\mathrm{m} := \mathsf{C} / \left(e(\mathsf{C}_0, \mathsf{K}_0) \cdot \prod_{i=1}^{\ell} e(\mathsf{C}_i, \mathsf{K}_i) \right) \in G_T$$

– $\mathsf{KeyDel}(\mathsf{PP}, \mathsf{SK}_{(\mathrm{id}_1,\ldots,\mathrm{id}_\ell)}, (\mathrm{id}'_1, \ldots, \mathrm{id}'_\ell))$ This algorithm picks $r'_1, \ldots, r'_{\ell'}, s'_1, \ldots, s'_{\ell'} \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and sets $s'_0 := s'_1 + \cdots + s'_{\ell'}$. The secret key is computed as

$$\mathsf{SK}_{(\mathrm{id}'_1,\ldots,\mathrm{id}'_{\ell'})} := \left\{ \mathsf{K}'_0 := \mathsf{K}_0 \cdot g_2^{-s'_0 \mathbf{d}^*_{1,0}}, \left\{ \mathsf{K}'_i := \mathsf{K}_i \cdot g_2^{s'_i \mathbf{d}^*_{1,i} + r'_i(\mathrm{id}'_i \mathbf{d}^*_{1,i} - \mathbf{d}^*_{2,i})} \right\}_{i=1,\ldots,\ell'} \right\} \in G_2^3 \times (G_2^4)^{\ell'}.$$

Correctness.

$$\begin{aligned} e(\mathsf{C}_0, \mathsf{K}_0) \cdot \prod_{i=1}^{\ell} e(\mathsf{C}_i, \mathsf{K}_i) &= g_T^{-s_0 z_0 + z} \cdot \prod_{i=1}^{\ell} g_T^{s_i z_0} \\ &= g_T^{z_0(-s_0 + \sum_{i=1}^{\ell} s_j) + z} = g_T^z. \end{aligned}$$