Simple and Efficient FE for Quadratic Functions

Junqing Gong and Haifeng Qian

East China Normal University
jqgong@sei.ecnu.edu.cn, hfqian@cs.ecnu.edu.cn

Abstract. This paper presents the first functional encryption schemes for quadratic functions (or degree-2 polynomials) achieving simulation-based security in the semi-adaptive model with *constant-size* secret key. The unique prior construction with the same security guarantee by Gay [PKC 20] has secret keys of size linear in the message size. They also enjoy shorter ciphertexts:

- our first scheme is based on bilateral DLIN (decisional linear) assumption as Gay's scheme and the ciphertext is 15% shorter;
- our second scheme based on SXDH assumption and bilateral DLIN assumption is more efficient; it has 67% shorter ciphertext than previous SXDH-based scheme with selective indistinguishability security by Baltico *et al.* [CRYPTO 17]; the efficiency is comparable to their second scheme in the generic group model.

Technically, we roughly combine Wee's "secret-key-to-public-key" compiler [TCC 17] with Gay's paradigm [PKC 20]. We avoid *(partial) function-hiding* inner-product functional encryption used in Gay's work and make our schemes conceptually simpler.

1 Introduction

Functional encryption (FE) [BSW11,O'N10] is an extension of attribute-based encryption (ABE) [SW05,GPSW06] which allows a user to recover partial information of encrypted data. More concretely, in a FE scheme for functionality $\mathcal{F}: \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, a ciphertext is associated with input $X \in \mathcal{X}$ while a key is associated with a function index $Y \in \mathcal{Y}$, decryption recovers $\mathcal{F}(X,Y) \in \mathcal{Z}$ without revealing any other information about X. There are two flavors of security, *indistinguishability-based security* (IND-security, for short) and *simulation-based security* (SIM-security, for short). The IND-security ensures that the adversary can not distinguish encryptions of two messages X_0, X_1 given a set of keys for $Y_1, \ldots, Y_Q$ such that $\mathcal{F}(X_0, Y_i) = \mathcal{F}(X_1, Y_i)$ for $i = 1, \ldots, Q$. The SIM-security ensures that an encryption of X and a set of keys $Y_1, \ldots, Y_Q$ can be simulated with $\mathcal{F}(X, Y_i)$ for all $i = 1, \ldots, Q$ instead of X. In general, SIM-security is stronger and more desirable in many theoretical applications. Actually, it is shown that, in the context of FE, IND-security is insufficient in some cases [BSW11,O'N10].

In practice, FE is useful in real-world applications requiring fine-grained access control like ABE; theoretically, FE for general functionality implies powerful primitives such as indistinguishability obfuscation [BV15,AJ15]. However such general FE have no construction from standard assumption such as k-Lin or LWE. In fact, existing constructions come from indistinguishability obfuscation/multilinear map [GGH13a,GGH$^+$13b,GGHZ16] with many candidates broken. Therefore, an important line of research is to build FE for concrete functionality whose security can be based on standard assumptions; this also provides us with reasonable efficiency suitable for real-world applications.

Abdalla *et al.* [ABDP15] proposed the first FE for inner product or linear function (IPFE): a ciphertext and a key are associated with $\mathbf{x} \in \mathbb{Z}_p^n$ and $\mathbf{y} \in \mathbb{Z}_p^n$, respectively; decryption recovers their inner product $\langle \mathbf{x}, \mathbf{y} \rangle$. Their proposals are based on DDH or LWE assumption but only achieve selective IND-security where the adversary is asked to commit the challenge before seeing public parameter. Agrawal *et al.* [ALS16] improved the result with adaptively IND-secure schemes from various standard assumptions even including DCR assumption; the challenge can be

chosen at any stage. It was also shown that SIM-security is achievable in the context of IPFE [Wee17,ALMT20] under various standard assumptions. This opens a fruitful research area studying extensions of IPFE such as multi-input/multi-client IPFE [AGRW17,ACF+18,CDG+18] supporting more complex scenarios. Theoretically, IPFE (with various features like function-hiding) also plays a crucial role in building other cryptographic primitives and sometimes offers a clean exposition [LL20].

Baltico *et al.* [BCFG17] proposed the first (public-key) functional encryption for quadratic function or degree-2 polynomial (QFFE): a ciphertext and a key are associated with $(\mathbf{x},\mathbf{y}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^m$ and $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$, respectively; decryption recovers $\mathbf{x}^\top\mathbf{F}\mathbf{y}$; it is additionally required that the ciphertext size is linear in $n+m$, i.e., the input size[1]. In their work, two schemes were proposed: one is selectively IND-secure based on SXDH and 3-party DDH assumption while another is adaptively IND-secure in the generic group model (GGM). Ryffel *et al.* [RPB+19,DGP18] described an efficient variant of Baltico *et al.*'s first scheme in GGM and showed its application in the field of privacy-preserving machine learning. There also exists several constructions [AS17,Lin17,JLS19] in the secret-key setting where the encryption procedure requires the knowledge of secret key. Recently, Gay [Gay20] proposed the first scheme achieving semi-adaptive SIM-security where the adversary can admit the challenge after seeing public parameter but before seeing any keys. The scheme is based on bilateral DLIN (decisional linear) assumption.

Motivation. As the unique SIM-secure scheme, Gay's work [Gay20] has a drawback in terms of efficiency — secret key size is linear in the size of message. We naturally pose the following question.

> *Q1: Can we achieve (semi-adaptive) SIM-security with shorter, say constant-size, secret keys?*

Given all known (IND-secure) QFFE schemes have constant-size keys, it is asking whether it is inevitable to blow up secret key in order to achieve SIM-security even in the weaker semi-adaptive model?

We also observe that there is an unexpected efficiency gap between existing QFFE schemes in GGM and in standard model. In the context of ABE, a selectively secure scheme in the standard model and an adaptively secure scheme in GGM typically have almost the same efficiency. However, in the context of QFFE, the schemes of the former type are less efficient; in particular, they roughly double the ciphertext size. This work also concerns the following question.

> *Q2: Can we achieve selective (or semi-adaptive) security with shorter ciphertexts, especially comparable to the schemes in GGM?*

Note that it is natural and reasonable to have larger ciphertexts in order to achieve adaptive security in the standard model such as those by dual-system method [Wat09,Wee14,Att14,CGW15,AC17b].

1.1 Results

In this paper, we affirmatively answer the aforementioned questions by proposing two QFFE schemes from standard assumptions in prime-order bilinear groups. Both of them

(1) achieve semi-adaptive SIM-security as in [Gay20];
(2) enjoy constant-size secret keys as all prior QFFE schemes except [Gay20];
(3) have shorter ciphertexts than prior QFFE schemes in the standard model;

and the second scheme is comparable to GGM-based schemes in [BCFG17, §4] and [RPB+19,DGP18] in terms of ciphertext size. We compare our schemes with existing schemes in Fig. 1:

[1] An IPFE scheme trivially implies a QFFE without the efficiency requirement, which is not interesting in many cases.

– The first scheme, denoted by Π_1, relies on the so-called *bilateral* k-Lin assumption. When we instantiate it with $k = 2$, i.e., bilateral DLIN assumption, the ciphertext in our scheme is 15% shorter than Gay's scheme based on the same assumption [Gay20] (roughly $6n$ vs. $7n$ for $|\mathbf{x}| = |\mathbf{y}| = n$).
– The second scheme, denoted by Π_2, is a variant of the first one; the security can be based on standard k-Lin assumption and bilateral d-Lin assumption while the ciphertext size is in the form $2(k+1)n + \mathsf{poly}(k,d)$. This allows us to get shorter ciphertext using $k = 1$, i.e., SXDH assumption as in [BCFG17, §3], the ciphertext is roughly 67% shorter than the scheme based on SXDH and 3PDDH in [BCFG17, §3] (roughly $4n$ vs. $12n$ for $|\mathbf{x}| = |\mathbf{y}| = n$).

reference	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	**security**	**assumption**
[BCFG17, §3]	$12n+2$	2	SEL-IND	SXDH + 3PDDH
[BCFG17, §4]	$4n+2$	2	AD-IND	GGM
[RPB+19,DGP18]	$4n+1$	1	AD-IND	GGM
[Gay20]	$7n+2$	$5n+2$	sAD-SIM	Bi-DLIN
Π_1 §4, §6.1	$6n+6$	6	sAD-SIM	Bi-DLIN
Π_2 §5, §6.2	$4n+10$	10	sAD-SIM	SXDH + Bi-DLIN

Fig. 1. Public-key functional encryption schemes for quadratic functions computing $\mathbf{x}^\top \mathbf{F}\mathbf{y}$ for $\mathbf{x} \in \mathbb{Z}_p^n$ and $\mathbf{y} \in \mathbb{Z}_p^m$. In the figure, we consider the case of $|\mathbf{x}| = |\mathbf{y}| = n$ and does not distinguish which source group contributes when counting the group elements in the ciphertext and secret key. In the column **security**, "SEL", "sAD" and "AD" stand for selective, semi-adaptive and adaptive model; "IND" and "SIM" stand for indistinguishably-based and simulation-based security. In the column assumption, "GGM" means generic group model.

A Quick Glance. To get our first QFFE scheme Π_1, we roughly follow the two-step workflow which is commonly used in building various functional encryptions [Wee17,CGW18,BCFG17]:

Step 1. We start from a secret-key QFFE which achieves selective SIM-security.
Step 2. We upgrade the secret-key scheme to public-key setting reaching semi-adaptive SIM-security.

In this work, we choose the secret-key scheme with constant-size keys in [BCFG17, §3] as the starting point in **Step 1**. We show that the scheme is actually selectively SIM-secure under k-Lin assumption; prior to our work, it is only known to be selectively IND-secure. Apart from this, we provide a compact and clean exposition of the scheme in the language of inner product of matrices which facilitates future adaptation. In **Step 2**, in order to upgrade the secret-key scheme we have in **Step 1** into the public-key setting, we integrate Wee's "secret-key-to-public-key" compiler [Wee17] with Gay's paradigm leading to his QFFE in [Gay20]. We indeed use IPFE as building block following [Gay20]; however, by borrowing the idea from Wee's compiler [Wee17], we avoid the reliance of partial function-hiding property used in [Gay20]. This preserves the constant-size keys and gives the first QFFE with semi-adaptive SIM-security with constant-size keys.

The scheme Π_1 relies on bilateral k-Lin assumption as in [Gay20], which does not hold for $k = 1$. With $k = 2$, the ciphertext size will be $6n$. We present an adaptation of Π_1, denoted by Π_2, which is motivated by [BCFG17] and employs the technique from multi-input IPFE (MIFE) [AGRW17]. Roughly, an idea shown in [BCFG17] allows us to replace bilateral k-Lin with standard k-Lin assumption where $k = 1$ is available; however we need a common

technique from MIFE to finalize the proof which introduces the use of bilateral d-Lin assumption again in our setting. However, this only causes an additive constant overhead to the ciphertext size and indeed gives us a scheme of ciphertext size $4n$, which is comparable to that in GGM.

1.2 Overview of Scheme Π_1

We give a technical overview of our first scheme Π_1 based on bilateral k-Lin assumption. Before that, we first introduce some notations. For $\mathbf{X} = (x_{ij}), \mathbf{Y} = (y_{ij}) \in \mathbb{Z}_p^{n\times m}$, we define their inner product as

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i,j} x_{ij} y_{ij} \in \mathbb{Z}_p.$$

This is a natural extension of inner product of vectors and is bilinear. One can compactly write it in the trace of matrix:

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \mathsf{tr}(\mathbf{X}^\top \mathbf{Y}) \in \mathbb{Z}_p \tag{1}$$

where $\mathsf{tr}(\mathbf{M})$ for $\mathbf{M}$ over $\mathbb{Z}_p$ is the sum of entries on the diagonal. Then we can rewrite the quadratic function using inner product of matrices:

$$\mathbf{x}^\top \mathbf{F} \mathbf{y} = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle \in \mathbb{Z}_p$$

where $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$ describes quadratic function and $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^m$ is the input.

Starting Point. We start from the secret-key QFFE described in [BCFG17]. Let (G_1, G_2, G_T) be an asymmetric bilinear groups of prime order p equipped with bilinear map e; we let g_1, g_2, g_T be random generators of G_1, G_2, G_T. We employ the implicit representation of group elements: for a matrix $\mathbf{M}$ over $\mathbb{Z}_p$, we write $[\mathbf{M}]_i := g_i^{\mathbf{M}}$ where $i = 1, 2, T$ and the exponentiation is carried out component-wise. We review the scheme in the language of inner product of matrices as follows. Note that, essentially, this does not change the scheme but makes further manipulation easier due to the compactness.

$$\begin{aligned} \mathsf{msk} &: \mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}, \mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k} \\ \mathsf{ct}_{\mathbf{x},\mathbf{y}} &: [(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V})\mathbf{M}]_2 \\ \mathsf{sk}_{\mathbf{F}} &: [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle]_T \end{aligned} \tag{2}$$

where $\mathbf{M}^*$ and $\mathbf{M}$ are picked uniformly over $\mathbb{Z}_p^{(k+1)\times(k+1)}$ satisfying the restriction $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$ and "$\|$" means the concatenation of two matrices. The decryption is done in three steps:

1. compute $[\mathbf{P}]_T$ from the two terms of $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ via bilinear map e where

$$\mathbf{P} = \Big(\overbrace{(\mathbf{x}\|\mathbf{U})\mathbf{M}^*}^{\text{1st term}} \Big) \cdot \Big(\overbrace{(\mathbf{y}\|\mathbf{V})\mathbf{M}}^{\text{2nd term}} \Big)^\top = (\mathbf{x}\|\mathbf{U})(\mathbf{y}\|\mathbf{V})^\top = \mathbf{x}\mathbf{y}^\top + \mathbf{U}\mathbf{V}^\top \tag{3}$$

   the second equality uses the fact that $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$; note that this step does not require the presence of any secret key;
2. compute $[Z]_T$ from $[\mathbf{P}]_T$ and $\mathsf{sk}_{\mathbf{F}}$ via group operation on G_T where

$$Z = \langle \mathbf{F}, \overbrace{\mathbf{P}}^{(3)} \rangle - \overbrace{\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle}^{\mathsf{sk}_{\mathbf{F}}} = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle$$

   the second equality uses the linearity of inner product (of matrices);
3. recover Z from $[Z]_T$ via brute-force discrete log algorithm; here we will assume that Z is always polynomially bounded [ABDP15,BCFG17].

In [BCFG17], it is proved to be selectively IND-secure under MDDH assumption which is implied by k-Lin assumption. However, by almost the same proof technique, we can show that the scheme is actually selectively SIM-secure under the same assumption. More concretely, we have

$$\begin{aligned} &\overbrace{[(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V})\mathbf{M}]_2}^{\text{real ct}}, \overbrace{[\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top\rangle]_T}^{\text{real sk}} \\ \approx_c\ &\underbrace{[\widetilde{\mathbf{U}}]_1, \quad [\widetilde{\mathbf{V}}]_2,}_{\text{simulated ct}} \underbrace{[\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top\rangle - \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top\rangle]_T}_{\text{simulated sk}} \end{aligned} \tag{4}$$

where $\widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}$ and $\widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}$ serves as the simulated master secret key. The second row shows the simulated ciphertext and key that solely leak $\langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top\rangle$ (as well as function $\mathbf{F}$).

From Secret-key to Public-key: Attempt & Issue. With a secret-key functional encryption scheme (2), we borrow the idea from Wee's "secret-key-to-public-key" compiler [Wee17] to get its public-key variant; the compiler is able to upgrade selectively secure secret-key scheme to semi-adaptively secure public-key scheme. (Note that our approach is not exactly the same as Wee's.) In particular, we will publish

$$\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k} \quad \text{and} \quad \mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$$

in the exponent as master public key and apply the following substitution to (2):

$$\mathbf{U} \longmapsto \mathbf{A}\mathbf{S} \quad \text{and} \quad \mathbf{V} \longmapsto \mathbf{B}\mathbf{T}$$

where $\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$. This yields the following public-key "scheme":

$$\begin{aligned} \mathsf{mpk} &: [\mathbf{A}]_1 \leftarrow G_1^{n\times k}, [\mathbf{B}]_2 \leftarrow G_2^{m\times k} \\ \mathsf{ct}_{\mathbf{x},\mathbf{y}} &: [(\mathbf{x}\|\mathbf{A}\mathbf{S})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2 \\ \mathsf{sk}_{\mathbf{F}} &: [\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_T \end{aligned} \tag{5}$$

Observe that, given $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ and $\mathsf{sk}_{\mathbf{F}}$ in this form, decryption works as before with $\mathbf{AS}, \mathbf{BT}$ in the place of $\mathbf{U}, \mathbf{V}$. Furthermore, with MDDH assumption:

$$[\mathbf{A}]_1, \boxed{[\mathbf{A}\mathbf{S}]_1} \approx_c [\mathbf{A}]_1, \boxed{[\mathbf{U}]_1} \quad \text{and} \quad [\mathbf{B}]_2, \boxed{[\mathbf{B}\mathbf{T}]_2} \approx_c [\mathbf{B}]_2, \boxed{[\mathbf{V}]_2}$$

we have the following argument which decouples both ciphertext and keys from mpk and reduces the "semi-adaptive security" of (5) to the selective security of (2):

$$\begin{aligned} &\overbrace{[\mathbf{A}]_1, [\mathbf{B}]_2}^{\mathsf{mpk}}, \overbrace{[(\mathbf{x}\|\boxed{\mathbf{A}\mathbf{S}})\mathbf{M}^*]_1, [(\mathbf{y}\|\boxed{\mathbf{B}\mathbf{T}})\mathbf{M}]_2}^{\mathsf{ct}_{\mathbf{x},\mathbf{y}}}, \overbrace{[\langle \mathbf{F}, \boxed{\mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top}\rangle]_T}^{\mathsf{sk}_{\mathbf{F}}} \\ \approx_c\ &[\mathbf{A}]_1, [\mathbf{B}]_2, \underbrace{[(\mathbf{x}\|\boxed{\mathbf{U}})\mathbf{M}^*]_1, \quad [(\mathbf{y}\|\boxed{\mathbf{V}})\mathbf{M}]_2, \quad [\langle \mathbf{F}, \boxed{\mathbf{U}\mathbf{V}^\top}\rangle]_T}_{\text{scheme (2) with } \mathsf{msk} = (\mathbf{U},\mathbf{V})} \end{aligned} \tag{6}$$

However the secret key and ciphertext in (5) will have to share random coins $\mathbf{S}$ and $\mathbf{T}$. In fact we may naturally view $\mathbf{S}$ and $\mathbf{T}$ as the random coins for ciphertext. Clearly, this violates the syntax of functional encryption: in order to decrypt a ciphertext with random coins $\mathbf{S}$ and $\mathbf{T}$, one might need a specific key with $\mathbf{S}$ and $\mathbf{T}$ embedded inside.

From Secret-key to Public-key: Solution. To fix the issue we take a closer look at the structure of secret key in (5) and observe that

$$\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle = \mathsf{tr}(\mathbf{F}^\top\mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top) = \mathsf{tr}(\mathbf{B}^\top\mathbf{F}^\top\mathbf{A}\mathbf{S}\mathbf{T}^\top) = \langle \mathbf{A}^\top\mathbf{F}\mathbf{B}, \mathbf{S}\mathbf{T}^\top\rangle. \tag{7}$$

Here the first and the last equalities follows the definition, cf.(1), while the second equality uses the property of trace: for matrices $\mathbf{A}, \mathbf{B}$ of the same size, it holds that $\mathsf{tr}(\mathbf{A}^\top\mathbf{B}) = \mathsf{tr}(\mathbf{B}\mathbf{A}^\top)$. Notice that, in last expression in (7),

– $\mathbf{A}^\top\mathbf{F}\mathbf{B}$ depends on mpk and quadratic function $\mathbf{F}$ and is independent of either random coins in the ciphertext or message $(\mathbf{x},\mathbf{y})$;
– $\mathbf{S}\mathbf{T}^\top$ solely depends on random coins in the ciphertext and is independent of quadratic function $\mathbf{F}$.

Therefore equation (7) suggests that we can fix the aforementioned issue if we can embed $\mathbf{A}^\top\mathbf{F}\mathbf{B}$ and $\mathbf{S}\mathbf{T}^\top$ into the key and the ciphertext (with random coins $\mathbf{S},\mathbf{T}$), respectively, such that one can recover $[\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_T$ during the decryption.

Our Scheme. We implement the above strategy using IPFE for matrices, denoted by $(\mathsf{KeyGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$, for inner product of matrices. Here we require Enc_1 to encrypt a matrix over G_1, say $[\mathbf{X}]_1$, and KeyGen_1 to generate a key for a matrix over G_2, say $[\mathbf{Y}]_2$, of the same dimension as $\mathbf{X}$. The decryption recovers the inner product of $\mathbf{X}$ and $\mathbf{Y}$ over G_T, i.e., $[\langle \mathbf{Y},\mathbf{X}\rangle]_T$. Keeping the observation (7) in mind, we

$$\text{split } [\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_T \text{ into } \mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2), \mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1)$$

and modify the scheme (5) by

– attaching $\mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1)$ to $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ (with random coins $\mathbf{S},\mathbf{T}$);
– taking $\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2)$ as the secret key $\mathsf{sk}_{\mathbf{F}}$.

This yields our first QFFE scheme Π_1:

$$\begin{aligned} \mathsf{mpk} &: [\mathbf{A}]_1 \leftarrow G_1^{n\times k}, [\mathbf{B}]_2 \leftarrow G_2^{m\times k}, \boxed{\mathsf{Enc}_1, \mathsf{Dec}_1} \\ \mathsf{ct}_{\mathbf{x},\mathbf{y}} &: [(\mathbf{x}\|\mathbf{A}\mathbf{S})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, \boxed{\mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1)} \\ \mathsf{sk}_{\mathbf{F}} &: \boxed{\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2)} \end{aligned} \tag{8}$$

where $\mathbf{S},\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$. We use dashed boxes to highlight differences with (5). During the decryption, one first assemble $\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2)$ and $\mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1)$ together by the decryption procedure of IPFE as follows:

$$\overbrace{[\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_T}^{\mathsf{sk}_{\mathbf{F}} \text{ in (5)}} = \mathsf{Dec}_1(\overbrace{\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2)}^{\mathsf{sk}_{\mathbf{F}} \text{ in (8)}}, \overbrace{\mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1)}^{\mathsf{ct}_{\mathbf{x},\mathbf{y}} \text{ in (8)}}) \tag{9}$$

and then use the recovered term as a secret key in (5). Clearly, this matches the syntax of functional encryption, i.e., each key can be used to decrypt any ciphertext. As a matter of fact, the method basically follows [Gay20]; however we will not require the underlying IPFE to be partially function-hiding as in [Gay20]. Instead, we will show that an IPFE over G_1 with standard security, whose key indeed reveals $[\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2$ in our setting, has been sufficient for the proof.

Proof Overview. More concretely, we require that the underlying IPFE achieves selective SIM-security, namely there exists simulator $(\widetilde{\mathsf{Enc}}_1, \widetilde{\mathsf{KeyGen}}_1)$ which, in our setting, ensures that

$$\begin{aligned} &\mathsf{Enc}_1, \mathsf{Dec}_1, \mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1), \mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2) \\ \approx_c\ &\mathsf{Enc}_1, \mathsf{Dec}_1, \widetilde{\mathsf{Enc}}_1(), \qquad \widetilde{\mathsf{KeyGen}}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, [\langle \mathbf{A}^\top\mathbf{F}\mathbf{B}, \mathbf{S}\mathbf{T}^\top\rangle]_2) \end{aligned} \tag{10}$$

Our proof roughly consists of three steps putting aforementioned ideas together in a reversed order:

1. assemble $\mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1)$ and $\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2)$ together by the SIM-security of underlying IPFE, namely apply (10);
2. decouple the challenge ciphertext and keys with mpk following (6);

3. apply the selective SIM-security of secret-key QFFE, namely use (4);

in more detail, our proof employs the following hybrid arguments, each of which corresponds to one step:

$$
\begin{array}{rll}
 & \overbrace{[\mathbf{A}]_1, [\mathbf{B}]_2}^{\mathsf{mpk}}, & \overbrace{[(\mathbf{x}\|\mathbf{AS})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \boxed{\mathsf{Enc}_1([\mathbf{ST}^\top]_1)}}^{\mathsf{ct}_{\mathbf{x},\mathbf{y}}}, & \overbrace{\boxed{\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{FB}]_2)}}^{\mathsf{sk}_{\mathbf{F}}} \\
\approx_c & [\mathbf{A}]_1, [\mathbf{B}]_2, & [(\mathbf{x}\|\mathbf{AS})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \boxed{\widetilde{\mathsf{Enc}_1}()}, & \boxed{\widetilde{\mathsf{KeyGen}_1}([\mathbf{A}^\top\mathbf{FB}]_2, [\langle \mathbf{F}, \mathbf{AST}^\top\mathbf{B}^\top\rangle]_2)} \\
\approx_c & [\mathbf{A}]_1, [\mathbf{B}]_2, & [(\mathbf{x}\|\boxed{\mathbf{U}})\mathbf{M}^*]_1, [(\mathbf{y}\|\boxed{\mathbf{V}})\mathbf{M}]_2, \widetilde{\mathsf{Enc}_1}(), & \widetilde{\mathsf{KeyGen}_1}([\mathbf{A}^\top\mathbf{FB}]_2, [\langle \mathbf{F}, \boxed{\mathbf{UV}^\top}\rangle]_2) \\
\approx_c & [\mathbf{A}]_1, [\mathbf{B}]_2, & \underbrace{\boxed{[\widetilde{\mathbf{U}}]_1}, \quad \boxed{[\widetilde{\mathbf{V}}]_2}, \quad \widetilde{\mathsf{Enc}_1}()}_{\text{simulated } \mathsf{ct}_{\mathbf{x},\mathbf{y}}}, & \underbrace{\widetilde{\mathsf{KeyGen}_1}([\mathbf{A}^\top\mathbf{FB}]_2, \boxed{[\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top\rangle - \langle \mathbf{F}, \mathbf{xy}^\top\rangle]_2})}_{\text{simulated } \mathsf{sk}_{\mathbf{F}}}
\end{array}
\tag{11}
$$

where the last row gives out the structure of simulated ciphertext and key. We note that, compared with (6), the second $\approx_c$ in (11) (cf. second bullet) should additionally take the leakage $\mathbf{A}^\top\mathbf{FB}$ (of master secret key $\mathbf{A}$ and $\mathbf{B}$) into account. In [Gay20], a similar leakage is handled by using *partially function-hiding* IPFE. However, observe that the term appears over G_2 in our scheme and this leaks no more information than mpk up to which source group is used. In fact, when we work with MDDH assumption (and its variant) as in our case, we do not need to worry about this kind of leakage at all. This almost works as is except we have to replace MDDH assumption w.r.t. $[\mathbf{A}]_1$ with its bilateral variant:

$$[\mathbf{A}]_1, [\mathbf{A}]_2, \boxed{[\mathbf{AS}]_1, [\mathbf{AS}]_2} \approx_c [\mathbf{A}]_1, [\mathbf{A}]_2, \boxed{[\mathbf{U}]_1, [\mathbf{U}]_2}$$

where $\mathbf{A}, \mathbf{AS}$ are also given out over G_2. The reasons are:

- In the second $\approx_c$ of (11), term $[\langle \mathbf{F}, \mathbf{AST}^\top\mathbf{B}^\top\rangle]_2$ and $[\mathbf{A}^\top\mathbf{FB}]_2$ (which involves $\mathbf{A}$) are over G_2 instead of G_T as in (6). We will need terms on G_2 in the assumption to simulate secrete key;
- In the third $\approx_c$ of (11), term $[\langle \mathbf{F}, \mathbf{UV}^\top\rangle]_2$ is over G_2 instead of G_T in scheme (4); the corresponding secret-key QFFE can be proved from bilateral MDDH assumption for the same reason as the first bullet.

1.3 More Efficient Scheme: Overview of Scheme Π_2

We show how to get our scheme Π_2 with more efficient instantiation.

Strategy. According our discussion in the end of last section, in order to avoid bilateral MDDH assumption, the simulator should have the ability to switch $\mathbf{A}^\top\mathbf{FB}$, $\langle \mathbf{F}, \mathbf{AST}^\top\mathbf{B}^\top\rangle$ and $\langle \mathbf{F}, \mathbf{UV}^\top\rangle$ between G_1 and G_2:

- when we use MDDH assumption w.r.t. $[\mathbf{A}]_1$, they appear over G_1;
- when we use MDDH assumption w.r.t. $[\mathbf{B}]_2$, they appear over G_2.

Let us focus on $\mathbf{A}^\top\mathbf{FB}$ which appears in the scheme and has the simplest structure among the three terms. The two requirements can be realized by an idea from [BCFG17]: we

$$\text{split } [\mathbf{A}^\top\mathbf{FB}]_2 \text{ into } [\mathbf{A}^\top\mathbf{FB} - \mathbf{R}]_2, [\mathbf{R}]_1$$

where $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times k}$ and the proof relies on the following statistical statement: for all $\mathbf{A}, \mathbf{B}, \mathbf{F}$, we have

$$[\mathbf{A}^\top\mathbf{FB} - \mathbf{R}]_2, [\mathbf{R}]_1 \approx_s [\mathbf{R}]_2, [\mathbf{A}^\top\mathbf{FB} - \mathbf{R}]_1$$

over the probability space defined by $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times k}$. The form on the left-hand side will be used when MDDH assumption w.r.t $[\mathbf{B}]_2$ is required; the form on the right-hand side will be used when MDDH assumption w.r.t. $[\mathbf{A}]_1$ is required.

Attempt. To handle the extra term $[\mathbf{R}]_1$ on the key side, we use IPFE for matrices over G_2, denoted $(\mathsf{KeyGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$; this is analogous to IPFE over G_1 except that we switch the role of G_1 and G_2: Enc_2 is used to encrypt $[\mathbf{X}]_2$ and KeyGen_2 is used to generate a key for $[\mathbf{Y}]_1$. In particular, we adapt Π_1, cf. (8), by

- setting $\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}-\mathbf{R}]_2)$, $\mathsf{KeyGen}_2([\mathbf{R}]_1)$ as $\mathsf{sk}_\mathbf{F}$;
- attaching $\mathsf{Enc}_2([\mathbf{S}\mathbf{T}^\top]_2)$ to the ciphertext $\mathsf{ct}_{\mathbf{x},\mathbf{y}}$ with random coins $\mathbf{S},\mathbf{T}$.

The first change corresponds to our strategy while the second one fixes the issue on correctness caused by the first change. This yields the following scheme:

$$\begin{aligned}\mathsf{mpk} &: [\mathbf{A}]_1 \leftarrow G_1^{n\times k}, [\mathbf{B}]_2 \leftarrow G_2^{m\times k}, \mathsf{Enc}_1, \mathsf{Dec}_1, \boxed{\mathsf{Enc}_2, \mathsf{Dec}_2}\\ \mathsf{ct}_{\mathbf{x},\mathbf{y}} &: [(\mathbf{x}\|\mathbf{A}\mathbf{S})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, \mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1), \boxed{\mathsf{Enc}_2([\mathbf{S}\mathbf{T}^\top]_2)}\\ \mathsf{sk}_\mathbf{F} &: \boxed{\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}-\mathbf{R}]_2), \mathsf{KeyGen}_2([\mathbf{R}]_1)}\end{aligned} \tag{12}$$

where $\mathbf{S},\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$. We use dashed boxes to highlight differences with our main Π_1, i.e.,(8). Decryption recovers

$$\begin{aligned}[\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle - \langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top\rangle]_T &= \mathsf{Dec}_1(\mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}-\mathbf{R}]_2), \mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1))\\ [\langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top\rangle]_T &= \mathsf{Dec}_2(\mathsf{KeyGen}_2([\mathbf{R}]_1), \mathsf{Enc}_2([\mathbf{S}\mathbf{T}^\top]_2))\end{aligned}$$

which are sufficient to compute $[\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_T$ as in (9) for the correctness.

Scheme. Although $\langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top\rangle$ indeed connects the two IPFE instances for the correctness but this is not sufficient for the proof. We employ the idea of connecting IPFE instances from multi-input IPFE [AGRW17]. For this, IPFE is extended to two-slot variant: ciphertext and key are additionally associated with a vector (over group) and decryption recovers the inner product of matrices (as usual) plus the inner product of vectors. Our second scheme Π_2 is as follows:

$$\begin{aligned}\mathsf{mpk} &: [\mathbf{A}]_1 \leftarrow G_1^{n\times k}, [\mathbf{B}]_2 \leftarrow G_2^{m\times k}, \mathsf{Enc}_1, \mathsf{Dec}_1, \mathsf{Enc}_2, \mathsf{Dec}_2\\ \mathsf{ct}_{\mathbf{x},\mathbf{y}} &: [(\mathbf{x}\|\mathbf{A}\mathbf{S})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, \mathsf{Enc}_1([\mathbf{S}\mathbf{T}^\top]_1, \boxed{[\mathbf{s}]_1}), \mathsf{Enc}_2([\mathbf{S}\mathbf{T}^\top]_2, \boxed{[\mathbf{s}]_2})\\ \mathsf{sk}_\mathbf{F} &: \mathsf{KeyGen}_1([\mathbf{A}^\top\mathbf{F}\mathbf{B}-\mathbf{R}]_2, \boxed{[-\mathbf{r}]_2}), \mathsf{KeyGen}_2([\mathbf{R}]_1, \boxed{[\mathbf{r}]_1})\end{aligned} \tag{13}$$

where $\mathbf{S},\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{s},\mathbf{r} \leftarrow \mathbb{Z}_p^d$. We use dashed boxes to highlight the extension on (12). Decryption procedure described above will recover

$$\begin{aligned}[\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle - \langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top\rangle - \boxed{\langle \mathbf{r}, \mathbf{s}\rangle}]_T\\ [\langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top\rangle + \boxed{\langle \mathbf{r}, \mathbf{s}\rangle}]_T\end{aligned}$$

The correctness is preserved. The proof will be analogous to (11) with an extra step extracting randomness τ from $\mathbf{r}$ and $\mathbf{s}$ following [AGRW17] (see H_2 in Section 5.3 and Lemma 5). This will give us the following terms involving $\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle$ and $\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top\rangle$ in the proof which have similar structure handling term $\mathbf{A}^\top\mathbf{F}\mathbf{B}$:

$$\begin{aligned}[\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle - \tau]_2,\ [\tau]_1,\\ [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top\rangle - \tau]_2,\ [\tau]_1.\end{aligned}$$

Note that this step itself requires bilateral MDDH assumption which increases the size of IPFE ciphertext; however this only causes constant overhead.

Concurrent Work. Gay *et al.* constructed a partially-hiding functional encryption (PHFE) to build iO [GJLS20]. Basically, their PHFE supports computation of quadratic functions where the function $\mathbf{F}$ is computed by a NC1 circuit (the circuit and input are associated with secret key and ciphertext, respectively). This definitively covers QFFE considered in our work. In fact, the derived QFFE scheme is quite similar to ours and the key size is constant; however it only achieves selective SIM-security as reported and has slightly larger ciphertexts.

Organization. We describe notations and definitions in Section 2. In Section 3, we revisit the secret-key QFFE schemes from [BCFG17] which will serve as our starting point. Our two QFFE schemes from distinct assumptions will be presented in Section 4 and 5. We show concrete schemes in Section 6.

2 Preliminaries

Notations. We denote by $s \leftarrow S$ the fact that s is picked uniformly from a finite set S. We use $\approx_s$ to denote two distributions being statistically indistinguishable, and $\approx_c$ to denote two distributions being computationally indistinguishable. We use lower case boldface to denote vectors and upper case boldcase to denote matrices. For a square matrix $\mathbf{M}$, we use $\mathsf{tr}(\mathbf{M})$ to denote it trace, i.e., the sum of entries on the diagonal. Throughout the paper, we use prime number p to denote the order of underlying groups.

Inner product of matrices. We define the inner product of $\mathbf{X}, \mathbf{Y} \in \mathbb{Z}_p^{n\times m}$ as

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \mathsf{tr}(\mathbf{X}^\top \mathbf{Y}).$$

Assume $\mathbf{X} = (x_{ij})$ and $\mathbf{Y} = (y_{ij})$, one can verify that $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i,j} x_{ij} y_{ij}$ that is a natural extension of inner product of vectors. Furthermore, one can also consider this as the inner product of two vectors of length nm induced from $\mathbf{X}, \mathbf{Y}$; in particular, we have

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \langle \mathsf{vec}(\mathbf{X}), \mathsf{vec}(\mathbf{Y}) \rangle \tag{14}$$

where $\mathsf{vec}(\mathbf{M})$ is the vector of length nm formed by piling columns of $\mathbf{M} \in \mathbb{Z}_p^{n\times m}$.

2.1 Functional Encryptions

Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be sets, we call $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ an functionality; this induces a family of functions mapping from $\mathcal{X}$ to $\mathcal{Z}$ indexed by $\mathcal{Y}$. Let p be a prime, we will use the following two concrete functionalities in the paper.

Inner product (Linear) function $\mathsf{IP}_{n,m}$. Let $\mathcal{X} = \mathbb{Z}_p^{n\times m}$, $\mathcal{Y} = \mathbb{Z}_p^{n\times m}$ and $\mathcal{Z} = \mathbb{Z}_p$ for some $n, m \in \mathbb{N}$, we define

$$\mathsf{IP}_{n,m} : (\mathbf{X}, \mathbf{Y}) \longmapsto \langle \mathbf{X}, \mathbf{Y} \rangle.$$

Quadratic function $\mathsf{QF}_{n,m}$. Let $\mathcal{X} = \mathbb{Z}_p^n \times \mathbb{Z}_p^m$, $\mathcal{Y} = \mathbb{Z}_p^{n\times m}$ and $\mathcal{Z} = \mathbb{Z}_p$ for some $n, m \in \mathbb{N}$, we define

$$\mathsf{QF}_{n,m} : \big((\mathbf{x}, \mathbf{y}), \mathbf{F}\big) \longmapsto \mathbf{x}^\top \mathbf{F} \mathbf{y}.$$

Note that we have $\mathbf{x}^\top \mathbf{F} \mathbf{y} = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle$ which will be used throughout the paper.

Algorithm. A functional encryption (FE) Π for functionality $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ consists of four PPT algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{F}) \to (\mathsf{mpk}, \mathsf{msk})$: The Setup algorithm takes security parameter 1^λ and functionality $\mathcal{F}$ as input, outputs master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{Enc}(\mathsf{mpk}, X \in \mathcal{X}) \to \mathsf{ct}_X$: The Enc algorithm takes master public key mpk and message $X \in \mathcal{X}$ as input, outputs a ciphertext ct_X.
- $\mathsf{KeyGen}(\mathsf{msk}, Y \in \mathcal{Y}) \to \mathsf{sk}_Y$: The KeyGen algorithm takes master secret key msk and function index $Y \in \mathcal{Y}$ as input, outputs a functional secret key sk_Y.
- $\mathsf{Dec}(\mathsf{ct}_X, \mathsf{sk}_Y) \to Z \in \mathcal{Z}$: The decryption algorithm takes a ciphertext ct_X and a functional secret key sk_Y as input, outputs $Z \in \mathcal{Z}$.

In this paper we use IPFE and QFFE to indicate FE for $\mathsf{IP}_{n,m}$ and $\mathsf{QF}_{n,m}$, respectively, for short.

Correctness. For all $\lambda \in \mathbb{N}$, $X \in \mathcal{X}$, $Y \in \mathcal{Y}$, we require that

$$\Pr\left[\mathsf{Dec}(\mathsf{ct}_X, \mathsf{sk}_Y) = \mathcal{F}(X,Y) : \begin{array}{l}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}) \\ \mathsf{ct}_X \leftarrow \mathsf{Enc}(\mathsf{mpk}, X) \\ \mathsf{sk}_Y \leftarrow \mathsf{KeyGen}(\mathsf{msk}, Y)\end{array}\right] = 1.$$

As a relaxation, we require the correctness described above holds when $\mathcal{F}(X,Y) \in B$ where $B \subseteq \mathbb{Z}_p$ has polynomial size. This comes from the use of discrete-log algorithm during decryption as in [ABDP15,ALS16].

Semi-adaptive simulation-based security (SIM-security). For every efficient stateful adversary $\mathcal{A}$, there exists simulator $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$ such that

$$\left[\begin{array}{l}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}); \\ X^* \leftarrow \mathcal{A}(\mathsf{mpk}); \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, X^*); \\ \textbf{output } \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}, \mathsf{ct}^*)\end{array}\right] \approx_c \left[\begin{array}{l}(\mathsf{mpk},\widetilde{\mathsf{msk}}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}); \\ X^* \leftarrow \mathcal{A}(\mathsf{mpk}); \\ \widetilde{\mathsf{ct}}^* \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}); \\ \textbf{output } \mathcal{A}^{\widetilde{\mathsf{KeyGen}}(\mathsf{msk}^*,\cdot,\cdot)}(\mathsf{mpk}, \widetilde{\mathsf{ct}}^*)\end{array}\right]$$

where the algorithm $\widetilde{\mathsf{KeyGen}}(\mathsf{msk}^*,\cdot,\cdot)$ gets Y along with $\mathcal{F}(X^*,Y)$ whenever $\mathcal{A}$ makes a query $Y \in \mathcal{Y}$ to $\mathsf{KeyGen}(\mathsf{msk},\cdot)$. We use $\mathsf{Adv}^{\Pi}_{\mathcal{A}}(\lambda)$ to denote the advantage in distinguishing the distributions.

Secret-key FE. We also consider the secret-key FE where the algorithm Setup solely outputs msk and algorithm Enc takes msk instead of mpk as input. Both correctness and semi-adaptive SIM-security (basically selective security due to the absence of mpk) can be formulated analogously.

2.2 Prime-order Bilinear Groups

A generator $\mathcal{G}$ takes as input a security parameter 1^λ and outputs a description $\mathbb{G} := (p, G_1, G_2, G_T, e)$, where p is a prime of $\Theta(\lambda)$ bits, G_1, G_2 and G_T are cyclic groups of order p, and $e : G_1 \times G_2 \to G_T$ is a non-degenerate bilinear map. The group operations in G_1, G_2, G_T and the bilinear map e are computable in deterministic polynomial time in λ. Let $g_1 \in G_1$, $g_2 \in G_2$ and $g_T = e(g_1, g_2) \in G_T$ be the respective generators. We employ the *implicit representation* of group elements: for a matrix $\mathbf{M}$ over $\mathbb{Z}_p$, we define $[\mathbf{M}]_i := g_i^{\mathbf{M}}$ for all $i \in \{1, 2, T\}$, where exponentiation is carried out component-wise. Also, given $[\mathbf{A}]_1, [\mathbf{B}]_2$, we let $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$ when the multiplication is well-defined.

Computational assumptions. We review the matrix Diffie-Hellman (MDDH) assumption [EHK+13] over G_1; the variant over G_2 can be defined analogously.

Assumption 1 ($\mathrm{MDDH}^d_{k,\ell}$ Assumption over G_1) *Let $\ell, k, d \in \mathbb{N}$. For all PPT adversaries $\mathcal{A}$, the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{MDDH}^d_{k,\ell}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, \boxed{[\mathbf{MS}]_1}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, \boxed{[\mathbf{U}]_1}) = 1]\right|$$

where $\mathbb{G} := (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{M} \leftarrow \mathbb{Z}_p^{\ell \times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k \times d}$ *and* $\mathbf{U} \leftarrow \mathbb{Z}_p^{\ell \times d}$.

Escala *et al.* [EHK+13] showed that it is tightly implied by k-Lin assumption. Note that the assumption unconditionally holds in the case $\ell \le k$. When $k = 1$, we call it symmetric external Diffie-Hellman (SXDH) assumption; when $k = 2$, we call it decisional linear (DLIN) assumption.

The bilateral matrix Diffie-Hellman (Bi-MDDH) assumption [AC17a,Gay20] extends the basic MDDH assumption by giving out $\mathbf{M}$ and $\mathbf{MS}$ (or $\mathbf{U}$) over both G_1 and G_2 and is not stronger than the version on symmetric bilinear groups [EHK+13], cf. [AC17a,Gay20].

Assumption 2 (Bilateral MDDH$^d_{k,\ell}$ Assumption) *Let $\ell, d \in \mathbb{N}$ and $k \geq 2$. For all PPT adversaries $\mathcal{A}$, the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\text{Bi-MDDH}^d_{k,\ell}}_{\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}(\mathbb{G}, \{[\mathbf{M}]_i, \boxed{[\mathbf{MS}]_i}\}_{i\in\{1,2\}}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, \{[\mathbf{M}]_i, \boxed{[\mathbf{U}]_i}\}_{i\in\{1,2\}}) = 1] \right|$$

where $\mathbb{G} := (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{M} \leftarrow \mathbb{Z}_p^{\ell\times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k\times d}$ *and* $\mathbf{U} \leftarrow \mathbb{Z}_p^{\ell\times d}$.

Note that it does not hold with parameter $k = 1$; with other paratmers, we have Bi-MDDH$^d_{k,\ell} \Rightarrow$ MDDH$^d_{k,\ell}$. When $k = 2$, we call it bilateral decisional linear (Bi-DLIN) assumption.

2.3 Building Block: (Two-slot) IPFE in Bilinear Groups

Let k and d be two independent parameters. We will use two-slot IPFE over G_1 which is a FE for the following functionality:

$$\mathcal{X} = G_1^{k\times k} \times G_1^d,\ \mathcal{Y} = G_2^{k\times k} \times G_2^d,\ \mathcal{Z} = G_T$$

$$\mathcal{F} : [\mathbf{X}, \mathbf{x}]_1 \times [\mathbf{Y}, \mathbf{y}]_2 \longmapsto [\langle \mathbf{X}, \mathbf{Y}\rangle + \langle \mathbf{x}, \mathbf{y}\rangle]_T$$

and equipped with $\widetilde{\mathsf{KeyGen}}$ taking an element from G_2 as the last input for the SIM-security. (Recall that the first input is $\widetilde{\mathsf{msk}}$ while the second input is in $\mathcal{Y} = G_2^{k\times k} \times G_2^d$ as defined). An IPFE over G_2 can be defined analogously by switching the role of G_1 and G_2. By (14), we can adapt Wee's IPFE scheme over cyclic groups [Wee17] to realize these two functionalities; we use IPFE_1 and IPFE_2 to denote the two schemes, respectively. Note that we always use the instance under SXDH assumption. When we invoke IPFE_1 and IPFE_2, we take 1^k and 1^d as inputs of Setup algorithm. Furthermore, when we omit the input 1^d, the scheme is just standard IPFE over G_1 or G_2 and the input correspond to the second slot is omitted in Enc and KeyGen.

3 Revisiting Baltico *et al.*'s Secret-key QFFE

In this section, we review Baltico *et al.*'s secret-key QFFE scheme [BCFG17]. We begin with a variant, denoted by π_1, which achieves selective SIM-security under Bi-MDDH assumption. The scheme in [BCFG17], denoted by π_2 in this work, is described based on π_1 which achieves the same security guarantee but from MDDH assumption. With the inner product of matrices, we give a simple and clean exposition. The proof of SIM-security basically follows that in [BCFG17], we only describe the main theorem along with the simulator and leave the proof in the appendix for completeness. Note that it is previously known to be selective IND-secure [BCFG17].

3.1 π_1: Secret-key QFFE from Bi-MDDH

The secret-key QFFE scheme π_1 for $\mathsf{QF}_{n,m}$ in prime-order bilinear groups is described as follows.

– $\mathsf{Setup}(1^\lambda, 1^n, 1^m)$: Sample

$$\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}, \quad \mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$$

and output

$$\mathsf{msk} = (\mathbf{U}, \mathbf{V}).$$

– $\mathsf{Enc}(\mathsf{msk}, (\mathbf{x}, \mathbf{y}))$: Let $\mathbf{x} \in \mathbb{Z}_p^n$ and $\mathbf{y} \in \mathbb{Z}_p^m$. Sample

$$(\mathbf{M}, \mathbf{M}^*) \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)} \times \mathbb{Z}_p^{(k+1)\times(k+1)}$$

such that $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Output

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = \left([(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V})\mathbf{M}]_2 \right) \in G_1^{n\times(k+1)} \times G_2^{m\times(k+1)}.$$

– $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$: Let $\mathbf{F} \in \mathbb{Z}_p^{m\times n}$. Output

$$\mathsf{sk}_{\mathbf{F}} = [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle]_2 \in G_2.$$

– $\mathsf{Dec}(\mathsf{ct}_{\mathbf{x},\mathbf{y}}, \mathsf{sk}_{\mathbf{F}})$: Parse

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = ([\mathbf{C}_1]_1, [\mathbf{C}_2]_2) \quad \text{and} \quad \mathsf{sk}_{\mathbf{F}} = [K]_2.$$

Compute

$$[\mathbf{P}]_T = e([\mathbf{C}_1]_1, [\mathbf{C}_2^\top]_2), \quad [D]_T = \langle \mathbf{F}, [\mathbf{P}]_T \rangle, \quad [Z]_T = [D]_T \cdot e([1]_1, [K]_2)^{-1}$$

and recover $Z \in \mathbb{Z}_p$ via brute-force DLOG. Note that $\langle \mathbf{F}, \cdot \rangle$ is a linear function and thus $[D]_T$ can be computed from $[\mathbf{P}]_T$ with the knowledge of $\mathbf{F}$.

This is the same as the scheme described in the TECHNICAL OVERVIEW in [BCFG17, Section 3] (also see (2)) except that $\mathsf{sk}_{\mathbf{F}}$ is over G_2 instead of G_T.

Correctness. For all $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{y} \in \mathbb{Z}_p^m$ and $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$, we have

$$\mathbf{P} = \mathbf{x}\mathbf{y}^\top + \mathbf{U}\mathbf{V}^\top \tag{15}$$

$$D = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle + \langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle \tag{16}$$

$$Z = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle \tag{17}$$

Here equality (15) follows from the fact:

$$\mathbf{C}_1\mathbf{C}_2^\top = ((\mathbf{x}\|\mathbf{U})\mathbf{M}^*)((\mathbf{y}\|\mathbf{V})\mathbf{M})^\top = (\mathbf{x}\|\mathbf{U})(\mathbf{y}\|\mathbf{V})^\top = \mathbf{x}\mathbf{y}^\top + \mathbf{U}\mathbf{V}^\top.$$

where the second step follows from the fact $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Equality (16) follows from (15) and the linearity of $\langle \mathbf{F}, \cdot \rangle$. The last equality is straightforward. This readily proves the correctness.

Simulation-based Security. We have the following theorem stating that scheme π_1 described above achieves selective SIM-security under Bi-MDDH assumption.

Theorem 1. *For all adversaries $\mathcal{A}$, there exist algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\pi_1}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\text{Bi-MDDH}_{k,n}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\text{MDDH}_{k,m}}_{\mathcal{B}_2}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A})$.

We describe the simulator which will be used in the rest of the paper.

– $\widetilde{\mathsf{Setup}}(1^\lambda, 1^n, 1^m)$: Sample

$$\widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}, \quad \widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}$$

and output

$$\widetilde{\mathsf{msk}} = (\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}).$$

– $\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})$: Output

$$\widetilde{\mathsf{ct}} = \left([\widetilde{\mathbf{U}}]_1, [\widetilde{\mathbf{V}}]_2\right).$$

– $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{F}, [\mu]_2)$: Output

$$\widetilde{\mathsf{sk}}_{\mathbf{F}} = [\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top \rangle - \mu]_2.$$

We sketch the proof in Section A (cf. [BCFG17]).

3.2 π_2: Secret-key Scheme from MDDH

The secret-key QFFE scheme π_2 for $\mathsf{QF}_{n,m}$ in prime-order bilinear groups has the same Setup and Enc algorithms as π_1 but KeyGen and Dec working as follows:

– $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$: Sample $\tau \leftarrow \mathbb{Z}_p$ and output

$$\mathsf{sk}_{\mathbf{F}} = \left([\tau]_1, [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle - \tau]_2 \right) \in G_1 \times G_2.$$

– $\mathsf{Dec}(\mathsf{ct}_{\mathbf{x},\mathbf{y}}, \mathsf{sk}_{\mathbf{F}})$: Parse

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = ([\mathbf{C}_1]_1, [\mathbf{C}_2]_2) \quad \text{and} \quad \mathsf{sk}_{\mathbf{F}} = ([K_1]_1, [K_2]_2).$$

Compute

$$[\mathbf{P}]_T = e([\mathbf{C}_1]_1, [\mathbf{C}_2^\top]_2), \quad [D]_T = \langle \mathbf{F}, [\mathbf{P}]_T \rangle$$

and

$$[Z]_T = [D]_T \cdot e([K_1]_1, [1]_2)^{-1} \cdot e([1]_1, [K_2]_2)^{-1}.$$

Recover $Z \in \mathbb{Z}_p$ from $[Z]_T$ via brute-force DLOG.

Correctness. The correctness can be verified as that for π_1. In fact we compute the same $\mathbf{P}, D, Z$ by the fact:

$$e([K_1]_1, [1]_2)^{-1} \cdot e([1]_1, [K_2]_2)^{-1} = [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle]_T^{-1} = e([1]_1, [K]_2)^{-1}.$$

Simulation-based Security. We have the following theorem stating that scheme π_2 described above achieves selective SIM-security under MDDH assumption.

Theorem 2. *For all adversaries $\mathcal{A}$, there exist algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\pi_2}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{MDDH}_{k,n}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathrm{MDDH}_{k,m}}_{\mathcal{B}_2}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A})$.

The simulator has the same $\widetilde{\mathsf{Setup}}$ and $\widetilde{\mathsf{Enc}}$ as π_1 but $\widetilde{\mathsf{KeyGen}}$ working as follows:

– $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{F}, [\mu]_2)$: Sample $\tau \leftarrow \mathbb{Z}_p$ and output

$$\widetilde{\mathsf{sk}}_{\mathbf{F}} = \left([\tau]_1, [\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top \rangle - \tau - \mu]_2 \right)$$

We sketch the proof in Section A (cf. [BCFG17]).

4 Π_1: Our QFFE from BI-MDDH$_k$

In this section, we show our first QFFE scheme Π_1 which achieves semi-adaptive SIM-security under BI-MDDH$_k$ assumption. The scheme is based on the secret-key QFFE scheme π_1 and use IPFE over G_1 as building block.

4.1 Scheme

Let $\mathsf{IPFE}_1 = (\mathsf{Setup}_1, \mathsf{Enc}_1, \mathsf{KeyGen}_1, \mathsf{Dec}_1)$ be an IPFE over G_1, cf. Section 2.3. Our QFFE scheme Π_1 based on π_1 in prime-order bilinear groups is described as follows.

– $\mathsf{Setup}(1^\lambda, 1^n, 1^m)$: Run

$$(\mathsf{mpk}_1, \mathsf{msk}_1) \leftarrow \mathsf{Setup}_1(1^\lambda, 1^k)$$

and sample

$$\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}.$$

Output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2\big) \quad \text{and} \quad \mathsf{msk} = \big(\mathsf{msk}_1, \mathbf{A}, \mathbf{B}\big).$$

– $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}, \mathbf{y}))$: Let $\mathbf{x} \in \mathbb{Z}_p^n$ and $\mathbf{y} \in \mathbb{Z}_p^m$. Sample

$$(\mathbf{M}, \mathbf{M}^*) \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)} \times \mathbb{Z}_p^{(k+1)\times(k+1)}$$

such that $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Sample

$$\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$$

and output

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = \big([(\mathbf{x}\|\mathbf{AS})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \mathsf{Enc}_1(\mathsf{mpk}_1, [\mathbf{ST}^\top]_1)\big).$$

– $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$: Let $\mathbf{F} \in \mathbb{Z}_p^{m\times n}$. Output

$$\mathsf{sk}_\mathbf{F} = \mathsf{KeyGen}_1(\mathsf{msk}_1, [\mathbf{A}^\top\mathbf{FB}]_2).$$

– $\mathsf{Dec}(\mathsf{ct}_{\mathbf{x},\mathbf{y}}, \mathsf{sk}_\mathbf{F})$: Parse

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = ([\mathbf{C}_1]_1, [\mathbf{C}_2]_2, \mathsf{ct}_1) \quad \text{and} \quad \mathsf{sk}_\mathbf{F} = \mathsf{sk}_1$$

and recover

$$[L]_T \leftarrow \mathsf{Dec}_1(\mathsf{ct}_1, \mathsf{sk}_1).$$

Compute

$$[\mathbf{P}]_T = e([\mathbf{C}_1]_1, [\mathbf{C}_2^\top]_2), \quad [D]_T = \langle \mathbf{F}, [\mathbf{P}]_T \rangle, \quad [Z]_T = [D - L]_T$$

and recover Z from $[Z]_T$ via brute-force DLOG.

Correctness. For all $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{y} \in \mathbb{Z}_p^m$ and $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$, we have

$$L = \langle \mathbf{F}, \mathbf{AST}^\top\mathbf{B}^\top \rangle \tag{18}$$

$$\mathbf{P} = \mathbf{xy}^\top + \mathbf{AST}^\top\mathbf{B}^\top \tag{19}$$

$$D = \langle \mathbf{F}, \mathbf{xy}^\top \rangle + \langle \mathbf{F}, \mathbf{AST}^\top\mathbf{B}^\top \rangle \tag{20}$$

$$Z = \langle \mathbf{F}, \mathbf{xy}^\top \rangle \tag{21}$$

Here equality (18) follows from the correctness of IPFE_1 and the fact:

$$\begin{aligned} \langle \mathbf{A}^\top\mathbf{FB}, \mathbf{ST}^\top \rangle &= \mathsf{tr}\big((\mathbf{B}^\top\mathbf{F}^\top\mathbf{A})(\mathbf{ST}^\top)\big) \\ &= \mathsf{tr}\big(\mathbf{F}^\top(\mathbf{AST}^\top\mathbf{B}^\top)\big) \\ &= \langle \mathbf{F}, \mathbf{AST}^\top\mathbf{B}^\top \rangle \end{aligned}$$

The first and third steps follow from the definition of inner product (cf. Section 2) and the second step uses the property of trace (i.e., $\mathsf{tr}(\mathbf{A}^\top\mathbf{B}) = \mathsf{tr}(\mathbf{BA}^\top)$ for matrices $\mathbf{A}, \mathbf{B}$ of the same size). All remaining equalities can be verified analogously as (15), (16) and (17), respectively, in Section 3.1 with $\mathbf{AS}$ and $\mathbf{BT}$ in the place of $\mathbf{U}$ and $\mathbf{V}$. This readily proves the correctness.

4.2 Simulator

Before we proceed to the security proof, we show the simulator for Π_1. Let $(\widetilde{\mathsf{Setup}}_1, \widetilde{\mathsf{Enc}}_1, \widetilde{\mathsf{KeyGen}}_1)$ be the simulator for IPFE_1, the simulator for Π_1 works as follows.

– $\widetilde{\mathsf{Setup}}(1^\lambda, 1^n, 1^m)$: Run

$$(\mathsf{mpk}_1, \widetilde{\mathsf{msk}}_1) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k)$$

and sample

$$\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}, \widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}, \widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}.$$

Output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2\big) \quad \text{and} \quad \widetilde{\mathsf{msk}} = \big(\widetilde{\mathsf{msk}}_1, \mathbf{A}, \mathbf{B}, \widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}\big).$$

– $\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})$: Output

$$\widetilde{\mathsf{ct}} = \big([\widetilde{\mathbf{U}}]_1, [\widetilde{\mathbf{V}}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)\big).$$

– $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}_1, \mathbf{F}, \mu)$: Output

$$\widetilde{\mathsf{sk}}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, [\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top\rangle - \mu]_2).$$

4.3 Security

We prove the following theorem stating that our QFFE scheme Π_1 achieves semi-adaptive SIM-security under Bi-MDDH assumption.

Theorem 3. *For all adversaries $\mathcal{A}$, there exist algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$ such that*

$$\mathsf{Adv}^{\Pi_1}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{IPFE}_1}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\text{Bi-MDDH}^k_{k,n}}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\text{MDDH}^k_{k,m}}_{\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{\pi_1}_{\mathcal{B}_4}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3), \mathsf{Time}(\mathcal{B}_4) \approx \mathsf{Time}(\mathcal{A})$.

Game Sequence. Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^m$ be the semi-adaptive challenge. We prove the theorem via the following game sequence.

$\underline{\mathsf{G}_0}$**:** Real game.
$\underline{\mathsf{G}_1}$**:** Identical to G_0 except that we run

$$(\mathsf{mpk}_1, \boxed{\widetilde{\mathsf{msk}}_1}) \leftarrow \boxed{\widetilde{\mathsf{Setup}}_1}(1^\lambda, 1^k)$$

and return $\mathsf{mpk} = \big(\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2\big)$ at the beginning of the game and the challenge ciphertext and secret key for $\mathbf{F}$ are as follows:

$$\mathsf{ct}^* = \Big([(\mathbf{x}\|\mathbf{A}\mathbf{S})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, \boxed{\widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)}\Big)$$
$$\mathsf{sk}_{\mathbf{F}} = \boxed{\widetilde{\mathsf{KeyGen}}_1}(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, [\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_2)$$

where $\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$. We claim that $\mathsf{G}_0 \approx_c \mathsf{G}_1$. This follows from the selective SIM-security of IPFE_1. See Lemma 1 for more details proof.

$\underline{\mathsf{G}_2}$**:** Identical to G_1 except that the challenge ciphertext and secret key for $\mathbf{F}$ are as follows:

$$\mathsf{ct}^* = \big([(\mathbf{x}\|\boxed{\mathbf{U}})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)\big)$$
$$\mathsf{sk}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{FB}]_2, [\langle \mathbf{F}, \boxed{\mathbf{U}}\mathbf{T}^\top\mathbf{B}^\top\rangle]_2)$$

where $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$. We claim that $\mathsf{G}_1 \approx_c \mathsf{G}_2$. This follows from $\text{BI-MDDH}_{k,n}^k$ assumption:

$$[\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{AS}]_1, [\mathbf{AS}]_2 \approx_c [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{U}]_1, [\mathbf{U}]_2$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$. In the reduction, we use $[\mathbf{A}]_1$ and $[\mathbf{AS}]_1$ (or $[\mathbf{U}]_1$) to simulate mpk and $\widetilde{\mathsf{ct}}^*$; we use $[\mathbf{A}]_2$ and $[\mathbf{AS}]_2$ (or $[\mathbf{U}]_2$) to simulate *all* $\mathsf{sk}_{\mathbf{F}}$. See Lemma 2 for more details.

$\underline{\mathsf{G}_3}$**:** Identical to G_2 except that the challenge ciphertext and secret key for $\mathbf{F}$ are as follows:

$$\mathsf{ct}^* = \big([(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\boxed{\mathbf{V}})\mathbf{M}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)\big)$$
$$\mathsf{sk}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{FB}]_2, [\langle \mathbf{F}, \mathbf{U}\boxed{\mathbf{V}^\top}\rangle]_2).$$

where $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$ and $\mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$. We claim that $\mathsf{G}_2 \approx_c \mathsf{G}_3$. This follows from the $\text{MDDH}_{k,m}^k$ assumption:

$$[\mathbf{B}]_2, [\mathbf{BT}]_2 \approx_c [\mathbf{B}]_2, [\mathbf{V}]_2$$

where $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$, $\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$. See Lemma 3 for more details.

$\underline{\mathsf{G}_4}$**:** Identical to G_3 except that the challenge ciphertext and secret key for $\mathbf{F}$ are as follows:

$$\widetilde{\mathsf{ct}}^* = \big(\boxed{[\widetilde{\mathbf{U}}]_1, [\widetilde{\mathbf{V}}]_2}, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)\big)$$
$$\widetilde{\mathsf{sk}}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{FB}]_2, \boxed{[\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top\rangle - \langle \mathbf{F}, \mathbf{xy}^\top\rangle]_2})$$

where $\widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}$ and $\widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}$. We claim that $\mathsf{G}_3 \approx_c \mathsf{G}_4$. This follows from the selective SIM-security of secret-key scheme π_1 under $\mathsf{msk} = (\mathbf{U}, \mathbf{V})$. See Lemma 4 for more details.

Note that G_4 can be simulated using the simulator described in Section 4.2 by setting $\mu = \langle \mathbf{F}, \mathbf{xy}^\top\rangle$.

4.4 Lemmas

Let $\mathsf{Adv}_i(\lambda)$ be the advantage function of $\mathcal{A}$ in G_i. We describe lemmas for $\mathsf{G}_{i-1} \approx_c \mathsf{G}_i$ with $i \in [4]$.

Lemma 1 **($\mathsf{G}_0 \approx_c \mathsf{G}_1$).** *There exists algorithm $\mathcal{B}_1$ such that* $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \le \mathsf{Adv}^{\mathsf{IPFE}_1}_{\mathcal{B}_1}(\lambda).$$

Proof (of Lemma 1). The lemma follows from the selective SIM-security of IPFE_1. In particular, the algorithm $\mathcal{B}_1$ works as follows:

(Setup) Pick $\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$. Submit $[\mathbf{ST}^\top]_1$ as the selective challenge of IPFE_1 and get back $(\mathsf{mpk}_1, \widehat{\mathsf{msk}}_1)$ along with $\widehat{\mathsf{ct}}_1^*$. Sample $\mathbf{A}, \mathbf{B}$ honestly and publish $\mathsf{mpk} = (\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2)$.

(Challenge) Once receiving the semi-adaptive challenge $(\mathbf{x}, \mathbf{y})$, sample $\mathbf{M}^*$ and $\mathbf{M}$ honestly and return

$$\widehat{\mathsf{ct}}^* = \big([(\mathbf{x}\|\mathbf{AS})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \widehat{\mathsf{ct}}_1^*\big)$$

where $\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{T}$ and $\widehat{\mathsf{ct}}_1^*$ are picked during **Setup**.

(Key Queries) On input $\mathbf{F}$, submit a key query $[\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2$ and return the response $\widehat{\mathsf{sk}}_{\mathbf{F}}$.

(Analysis) Observe that

- when $(\mathsf{mpk}_1, \widehat{\mathsf{msk}_1}) = (\mathsf{mpk}_1, \mathsf{msk}_1) \leftarrow \mathsf{Setup}_1(1^\lambda, 1^k)$ and
$$\widehat{\mathsf{ct}}_1^* \leftarrow \mathsf{Enc}_1(\mathsf{mpk}_1, [\mathbf{S}\mathbf{T}^\top]_1), \quad \widehat{\mathsf{sk}}_{\mathbf{F}} \leftarrow \mathsf{KeyGen}_1(\mathsf{msk}_1, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2)$$
the simulation is identical to G_0;
- when $(\mathsf{mpk}_1, \widehat{\mathsf{msk}_1}) = (\mathsf{mpk}_1, \widetilde{\mathsf{msk}_1}) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k)$ and
$$\widehat{\mathsf{ct}}_1^* \leftarrow \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}_1}), \quad \widehat{\mathsf{sk}}_{\mathbf{F}} \leftarrow \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}_1}, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, [\langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top\mathbf{B}^\top\rangle]_2)$$
the simulation is identical to G_1.

This readily proves the lemma. □

Lemma 2 ($\mathsf{G}_1 \approx_c \mathsf{G}_2$)**.** *There exists algorithm* $\mathcal{B}_2$ *such that* $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$|\mathsf{Adv}_1(\lambda) - \mathsf{Adv}_2(\lambda)| \le \mathsf{Adv}^{\text{BI-MDDH}^k_{k,n}}_{\mathcal{B}_2}(\lambda).$$

Proof (of Lemma 2). This follows from $\text{BI-MDDH}^k_{k,n}$ assumption:

$$[\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{A}\mathbf{S}]_1, [\mathbf{A}\mathbf{S}]_2 \approx_c [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{U}]_1, [\mathbf{U}]_2$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$. On input $[\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{Z}]_1, [\mathbf{Z}]_2$ where $\mathbf{Z} = \mathbf{A}\mathbf{S}$ or $\mathbf{Z} \leftarrow \mathbb{Z}_p^{n\times k}$, the algorithm $\mathcal{B}_2$ works as follows:

(Setup) Run $(\mathsf{mpk}_1, \widetilde{\mathsf{msk}_1}) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k)$, sample $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$ and output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2\big)$$

where $[\mathbf{A}]_1$ is fetched from the input.

(Challenge) Once receiving the semi-adaptive challenge $(\mathbf{x}, \mathbf{y})$, sample $\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{M}^*, \mathbf{M} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$ with $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Output

$$\widehat{\mathsf{ct}}^* = \big([(\mathbf{x}\|\mathbf{Z})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}_1})\big)$$

using $[\mathbf{Z}]_1$ from the input.

(Key Queries) On input $\mathbf{F}$, return

$$\widehat{\mathsf{sk}}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}_1}, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, [\langle \mathbf{F}, \mathbf{Z}\mathbf{T}^\top\mathbf{B}^\top\rangle]_2)$$

where $[\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2$ and $[\langle \mathbf{F}, \mathbf{Z}\mathbf{T}^\top\mathbf{B}^\top\rangle]_2$ are simulated using $[\mathbf{A}]_2$ and $[\mathbf{Z}]_2$, respectively. Note that $\mathbf{B}, \mathbf{T}$ are known.

(Analysis) Observe that, when $\mathbf{Z} = \mathbf{A}\mathbf{S}$, the simulation is identical to G_1; when $\mathbf{Z} \leftarrow \mathbb{Z}_p^{n\times k}$, the simulation is identical to G_2. This readily proves the lemma. □

Lemma 3 ($\mathsf{G}_2 \approx_c \mathsf{G}_3$)**.** *There exists algorithm* $\mathcal{B}_3$ *such that* $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$|\mathsf{Adv}_2(\lambda) - \mathsf{Adv}_3(\lambda)| \le \mathsf{Adv}^{\text{MDDH}^k_{k,m}}_{\mathcal{B}_3}(\lambda).$$

Proof (of Lemma 3). This follows from $\text{MDDH}^k_{k,m}$ assumption:

$$[\mathbf{B}]_2, [\mathbf{B}\mathbf{T}]_2 \approx_c [\mathbf{B}]_2, [\mathbf{V}]_2$$

where $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$, $\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$. On input $[\mathbf{B}]_2, [\mathbf{Z}]_2$ where $\mathbf{Z} = \mathbf{B}\mathbf{T}$ or $\mathbf{Z} \leftarrow \mathbb{Z}_p^{m\times k}$, the algorithm $\mathcal{B}_3$ works as follows:

(Setup) Run $(\mathsf{mpk}_1, \widetilde{\mathsf{msk}}_1) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k)$, sample $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$ and output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2\big)$$

where $[\mathbf{B}]_2$ is fetched from the input. Also sample $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$.

(Challenge) Once receiving the semi-adaptive challenge $(\mathbf{x}, \mathbf{y})$, sample $\mathbf{M}^*, \mathbf{M} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$ with $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Output

$$\widehat{\mathsf{ct}}^* = \big([(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{Z})\mathbf{M}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)\big)$$

using $[\mathbf{Z}]_2$ from the input.

(Key Queries) On input $\mathbf{F}$, return

$$\widehat{\mathsf{sk}}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, [\langle \mathbf{F}, \mathbf{U}\mathbf{Z}^\top\rangle]_2)$$

where $[\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2$ and $[\langle \mathbf{F}, \mathbf{U}\mathbf{Z}^\top\rangle]_2$ are simulated using $[\mathbf{B}]_2$ and $[\mathbf{Z}]_2$, respectively. Note that $\mathbf{A}, \mathbf{U}$ are known.

(Analysis) Observe that, when $\mathbf{Z} = \mathbf{B}\mathbf{T}$, the simulation is identical to G_2; when $\mathbf{Z} \leftarrow \mathbb{Z}_p^{m\times k}$, the simulation is identical to G_3. This readily proves the lemma. □

Lemma 4 **($\mathsf{G}_3 \approx_c \mathsf{G}_4$).** *There exists algorithm $\mathcal{B}_4$ such that* $\mathsf{Time}(\mathcal{B}_4) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$|\mathsf{Adv}_3(\lambda) - \mathsf{Adv}_4(\lambda)| \le \mathsf{Adv}^{\pi_1}_{\mathcal{B}_4}(\lambda).$$

Proof (of Lemma 4). The lemma follows from the selective SIM-security of π_1. In particular, the algorithm $\mathcal{B}_4$ works as follows:

(Setup) Run $(\mathsf{mpk}_1, \widetilde{\mathsf{msk}}_1) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k)$ and sample $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$. Output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, [\mathbf{A}]_1, [\mathbf{B}]_2\big).$$

(Challenge) Once receiving the semi-adaptive challenge $(\mathbf{x}, \mathbf{y})$, submit it as the selective challenge of π_1 and get back $\widehat{\mathsf{ct}}^*_{\pi_1}$. Return

$$\widehat{\mathsf{ct}}^* = \big(\widehat{\mathsf{ct}}^*_{\pi_1}, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1)\big).$$

(Key Queries) On input $\mathbf{F}$, submit a key query $\mathbf{F}$ and get the response $\widehat{\mathsf{sk}}_{\mathbf{F},\pi_1} \in G_2$. Output

$$\widehat{\mathsf{sk}}_{\mathbf{F}} = \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, [\mathbf{A}^\top\mathbf{F}\mathbf{B}]_2, \widehat{\mathsf{sk}}_{\mathbf{F},\pi_1})$$

where $\mathbf{A}, \mathbf{B}$ are picked during **Setup**.

(Analysis) Observe that

- when $\widehat{\mathsf{ct}}^*_{\pi_1} = ([(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V})\mathbf{M}]_2)$ and $\widehat{\mathsf{sk}}_{\mathbf{F},\pi_1} = [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top\rangle]_2$ with $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}, \mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$ and $\mathbf{M}^*, \mathbf{M} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$ with $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$, the simulation is identical to G_3;
- when when $\widehat{\mathsf{ct}}^*_{\pi_1} = ([\widetilde{\mathbf{U}}]_1, [\widetilde{\mathbf{V}}]_2)$ and $\widehat{\mathsf{sk}}_{\mathbf{F},\pi_1} = [\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top\rangle - \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top\rangle]_2$ with $\widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}$, $\widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}$, the simulation is identical to G_4.

This readily proves the lemma. □

5 Π_2: Our QFFE from MDDH_k and $\mathrm{BI\text{-}MDDH}_d$

In this section, we present our second QFFE scheme Π_2 based on Π_1. The scheme achieves semi-adaptive SIM-security as Π_1 but under standard MDDH_k and $\mathrm{BI\text{-}MDDH}_d$ assumption. Technically, we will additionally use IPFE over G_2 as building block and require two-slot extension; the underlying secret-key QFFE is π_2 instead of π_1.

5.1 Scheme

Let $\mathsf{IPFE}_1 = (\mathsf{Setup}_1, \mathsf{Enc}_1, \mathsf{KeyGen}_1, \mathsf{Dec}_1)$ be two-slot IPFE over G_1; $\mathsf{IPFE}_2 = (\mathsf{Setup}_2, \mathsf{Enc}_2, \mathsf{KeyGen}_2, \mathsf{Dec}_2)$ be two-slot IPFE over G_2, cf. Section 2.3. Our QFFE scheme Π_2 in prime-order bilinear groups is described as follows.

– $\mathsf{Setup}(1^\lambda, 1^n, 1^m)$: Run

$$(\mathsf{mpk}_1, \mathsf{msk}_1) \leftarrow \mathsf{Setup}_1(1^\lambda, 1^k, 1^d), (\mathsf{mpk}_2, \mathsf{msk}_2) \leftarrow \mathsf{Setup}_2(1^\lambda, 1^k, 1^d)$$

and sample

$$\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}.$$

Output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, \mathsf{mpk}_2, [\mathbf{A}]_1, [\mathbf{B}]_2\big) \quad \text{and} \quad \mathsf{msk} = \big(\mathsf{msk}_1, \mathsf{msk}_2, \mathbf{A}, \mathbf{B}\big).$$

– $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}, \mathbf{y}))$: Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^m$. Sample

$$(\mathbf{M}, \mathbf{M}^*) \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)} \times \mathbb{Z}_p^{(k+1)\times(k+1)}$$

such that $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Sample

$$\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}, \quad \mathbf{s} \leftarrow \mathbb{Z}_p^d$$

and output

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = \begin{pmatrix} [(\mathbf{x}\|\mathbf{AS})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \\ \mathsf{Enc}_1(\mathsf{mpk}_1, ([\mathbf{ST}^\top]_1, [\mathbf{s}]_1)), \mathsf{Enc}_2(\mathsf{mpk}_2, ([\mathbf{ST}^\top]_2, [\mathbf{s}]_2)) \end{pmatrix}.$$

– $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$: Let $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$. Sample $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{r} \leftarrow \mathbb{Z}_p^d$ and output

$$\mathsf{sk}_{\mathbf{F}} = \big(\mathsf{KeyGen}_1(\mathsf{msk}_1, ([\mathbf{A}^\top\mathbf{FB} - \mathbf{R}]_2, [-\mathbf{r}]_2)), \mathsf{KeyGen}_2(\mathsf{msk}_2, ([\mathbf{R}]_1, [\mathbf{r}]_1))\big).$$

– $\mathsf{Dec}(\mathsf{ct}_{\mathbf{x},\mathbf{y}}, \mathsf{sk}_{\mathbf{F}})$: Parse

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = ([\mathbf{C}_1]_1, [\mathbf{C}_2]_2, \mathsf{ct}_1, \mathsf{ct}_2) \quad \text{and} \quad \mathsf{sk}_{\mathbf{F}} = (\mathsf{sk}_1, \mathsf{sk}_2)$$

and recover

$$[L_1]_T \leftarrow \mathsf{Dec}_1(\mathsf{ct}_1, \mathsf{sk}_1), \quad [L_2]_T \leftarrow \mathsf{Dec}_2(\mathsf{ct}_2, \mathsf{sk}_2).$$

Compute

$$[\mathbf{P}]_T = e([\mathbf{C}_1]_1, [\mathbf{C}_2^\top]_2), \quad [D]_T = \langle \mathbf{F}, [\mathbf{P}]_T \rangle, \quad [Z]_T = [D - L_1 - L_2]_T$$

and recover Z from $[Z]_T$ via brute-force DLOG.

Correctness. For all $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{y} \in \mathbb{Z}_p^m$ and $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$, we have

$$L_1 = \langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \rangle - \langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top \rangle - \langle \mathbf{r}, \mathbf{s} \rangle \tag{22}$$

$$L_2 = \langle \mathbf{R}, \mathbf{S}\mathbf{T}^\top \rangle + \langle \mathbf{r}, \mathbf{s} \rangle \tag{23}$$

$$\mathbf{P} = \mathbf{x}\mathbf{y}^\top + \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \tag{24}$$

$$D = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle + \langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \rangle \tag{25}$$

$$Z = \langle \mathbf{F}, \mathbf{x}\mathbf{y}^\top \rangle \tag{26}$$

Here (23) follows the correctness of IPFE_2; the remaining four can be verified as in Section 4.1. In fact we compute the same $\mathbf{P}, D, Z$ by the fact that

$$L_1 + L_2 = \langle \mathbf{F}, \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \rangle = L.$$

This readily proves the correctness.

5.2 Simulator

Let $(\widetilde{\mathsf{Setup}}_1, \widetilde{\mathsf{Enc}}_1, \widetilde{\mathsf{KeyGen}}_1)$ and $(\widetilde{\mathsf{Setup}}_2, \widetilde{\mathsf{Enc}}_2, \widetilde{\mathsf{KeyGen}}_2)$ be the simulators of IPFE_1 and IPFE_2, respectively, the simulator for Π_2 is described as follows.

– $\widetilde{\mathsf{Setup}}(1^\lambda, 1^n, 1^m)$: Run

$$(\mathsf{mpk}_1, \widetilde{\mathsf{msk}}_1) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k, 1^d), (\mathsf{mpk}_2, \widetilde{\mathsf{msk}}_2) \leftarrow \widetilde{\mathsf{Setup}}_2(1^\lambda, 1^k, 1^d)$$

and sample

$$\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}, \widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}, \widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}.$$

Output

$$\mathsf{mpk} = \left(\mathsf{mpk}_1, \mathsf{mpk}_2, [\mathbf{A}]_1, [\mathbf{B}]_2\right) \quad \text{and} \quad \widetilde{\mathsf{msk}} = \left(\widetilde{\mathsf{msk}}_1, \widetilde{\mathsf{msk}}_2, \mathbf{A}, \mathbf{B}, \widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}\right)$$

– $\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})$: Output

$$\mathsf{ct} = \left([\widetilde{\mathbf{U}}]_1, [\widetilde{\mathbf{V}}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1), \widetilde{\mathsf{Enc}}_2(\widetilde{\mathsf{msk}}_2)\right).$$

– $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{F}, \mu)$: Sample $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{r} \leftarrow \mathbb{Z}_p^d$, $\tau \leftarrow \mathbb{Z}_p$ and output

$$\mathsf{sk}_\mathbf{F} = \begin{pmatrix} \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, ([\mathbf{A}^\top \mathbf{F}\mathbf{B} - \mathbf{R}]_2, [-\mathbf{r}]_2), [\langle \mathbf{F}, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top \rangle - \tau - \mu]_2) \\ \widetilde{\mathsf{KeyGen}}_2(\widetilde{\mathsf{msk}}_2, ([\mathbf{R}]_1, [\mathbf{r}]_1), [\tau]_1) \end{pmatrix}.$$

5.3 Security

We prove the following theorem stating that our QFFE scheme Π_2 achieves semi-adaptive SIM-security from MDDH_k and $\mathrm{BI\text{-}MDDH}_d$ assumption.

Theorem 4. *For all adversaries $\mathcal{A}$, there exist algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6$ such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\Pi_2}(\lambda) \le \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{IPFE}_1}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{IPFE}_2}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathrm{BI\text{-}MDDH}_{d,Q}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_4}^{\mathrm{MDDH}_{k,n}^k}(\lambda) + \mathsf{Adv}_{\mathcal{B}_5}^{\mathrm{MDDH}_{k,m}^k}(\lambda) + \mathsf{Adv}_{\mathcal{B}_6}^{\pi_2}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3), \mathsf{Time}(\mathcal{B}_4), \mathsf{Time}(\mathcal{B}_5), \mathsf{Time}(\mathcal{B}_6) \approx \mathsf{Time}(\mathcal{A})$ *where Q is the number of key queries.*

Game Sequence. Let $(\mathbf{x},\mathbf{y}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^m$ be the semi-adaptive challenge. We prove the theorem via the following game sequence. WLOG, we assume the adversary $\mathcal{A}$ makes Q key queries $\mathbf{F}_1,\dots,\mathbf{F}_Q$ and we will use specific subscript to indicate random coins used in each key.

$\underline{\mathsf{H}_0}$**:** Real game.

$\underline{\mathsf{H}_1}$**:** Identical to H_0 except that we run

$$(\mathsf{mpk}_1, \boxed{\widetilde{\mathsf{msk}_1}}) \leftarrow \boxed{\widetilde{\mathsf{Setup}_1}}(1^\lambda, 1^k, 1^d), (\mathsf{mpk}_2, \boxed{\widetilde{\mathsf{msk}_2}}) \leftarrow \boxed{\widetilde{\mathsf{Setup}_2}}(1^\lambda, 1^k, 1^d)$$

and return $\mathsf{mpk} = \big(\mathsf{mpk}_1, \mathsf{mpk}_2, [\mathbf{A}]_1, [\mathbf{B}]_2\big)$ at the beginning of the game and the challenge ciphertext and secret key for $\mathbf{F}_i$ with $i \in [Q]$ are as follows:

$$\begin{aligned}
\mathsf{ct}^* &= \Big([(\mathbf{x}\|\mathbf{AS})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \boxed{\widetilde{\mathsf{Enc}_1}(\widetilde{\mathsf{msk}_1})}, \boxed{\widetilde{\mathsf{Enc}_2}(\widetilde{\mathsf{msk}_2})}\Big) \\
\mathsf{sk}_{\mathbf{F}_i} &= \begin{pmatrix} \boxed{\widetilde{\mathsf{KeyGen}_1}}\Big(\widetilde{\mathsf{msk}_1}, ([\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [-\mathbf{r}_i]_2), [\langle \mathbf{F}_i, \mathbf{AST}^\top \mathbf{B}^\top\rangle - \langle \mathbf{R}_i, \mathbf{ST}^\top\rangle - \langle \mathbf{r}_i, \mathbf{s}\rangle]_2\Big) \\ \boxed{\widetilde{\mathsf{KeyGen}_2}}(\widetilde{\mathsf{msk}_2}, ([\mathbf{R}_i]_1, [\mathbf{r}_i]_1), [\langle \mathbf{R}_i, \mathbf{ST}^\top\rangle + \langle \mathbf{r}_i, \mathbf{s}\rangle]_1) \end{pmatrix}
\end{aligned}$$

where $\mathbf{S},\mathbf{T},\mathbf{R}_i \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{r}_i \leftarrow \mathbb{Z}_p^d$ for all $i \in [Q]$. We claim that $\mathsf{H}_0 \approx_c \mathsf{H}_1$. This follows from the selective SIM-security of IPFE_1 and IPFE_2. This is analogous to "$\mathsf{G}_0 \approx_c \mathsf{G}_1$" for Π_1 in Section 4.3. We omit the detail.

$\underline{\mathsf{H}_2}$**:** Identical to H_1 except that the challenge ciphertext and secret key for $\mathbf{F}_i$ are as follows:

$$\mathsf{sk}_{\mathbf{F}_i} = \begin{pmatrix} \widetilde{\mathsf{KeyGen}_1}(\widetilde{\mathsf{msk}_1}, ([\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [-\mathbf{r}_i]_2), [\langle \mathbf{F}_i, \mathbf{AST}^\top \mathbf{B}^\top\rangle - \boxed{\tau_i}]_2) \\ \widetilde{\mathsf{KeyGen}_2}(\widetilde{\mathsf{msk}_2}, ([\mathbf{R}_i]_1, [\mathbf{r}_i]_1), \boxed{[\tau_i]_1}) \end{pmatrix}$$

where $\tau_i \leftarrow \mathbb{Z}_p$ for all $i \in [Q]$. We claim the $\mathsf{H}_1 \approx_c \mathsf{H}_2$. This follows from the $\text{BI-MDDH}_{d,Q}$ assumption which implies that, for all $\mathbf{S},\mathbf{T},\mathbf{R}_i$, we have

$$\begin{aligned}
&\{[\mathbf{r}_i]_1, [\mathbf{r}_i]_2, [\langle \mathbf{R}_i, \mathbf{ST}^\top\rangle + \langle \mathbf{r}_i, \mathbf{s}\rangle]_1, [\langle \mathbf{R}_i, \mathbf{ST}^\top\rangle + \langle \mathbf{r}_i, \mathbf{s}\rangle]_2\}_{i\in[Q]} \\
\approx_c\ &\{[\mathbf{r}_i]_1, [\mathbf{r}_i]_2, [\tau_i]_1, \qquad\qquad [\tau_i]_2 \qquad\qquad\}_{i\in[Q]}
\end{aligned}$$

where $\mathbf{r}_i, \mathbf{s} \leftarrow \mathbb{Z}_p^d$ and $\tau_i \leftarrow \mathbb{Z}_p$ for all $i \in [Q]$. See Lemma 5 for more details.

$\underline{\mathsf{H}_3}$**:** Identical to H_2 except that the challenge ciphertext and secret key for $\mathbf{F}_i$ are as follows:

$$\begin{aligned}
\mathsf{ct}^* &= \Big([(\mathbf{x}\|\boxed{\mathbf{U}})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{BT})\mathbf{M}]_2, \widetilde{\mathsf{Enc}_1}(\widetilde{\mathsf{msk}_1}), \widetilde{\mathsf{Enc}_2}(\widetilde{\mathsf{msk}_2})\Big) \\
\mathsf{sk}_{\mathbf{F}_i} &= \begin{pmatrix} \widetilde{\mathsf{KeyGen}_1}(\widetilde{\mathsf{msk}_1}, ([\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [-\mathbf{r}_i]_2), [\langle \mathbf{F}_i, \boxed{\mathbf{U}}\mathbf{T}^\top \mathbf{B}^\top\rangle - \tau_i]_2) \\ \widetilde{\mathsf{KeyGen}_2}(\widetilde{\mathsf{msk}_2}, ([\mathbf{R}_i]_1, [\mathbf{r}_i]_1), [\tau_i]_1) \end{pmatrix}
\end{aligned}$$

where $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$. We claim that $\mathsf{H}_2 \approx_c \mathsf{H}_3$. This follows from the $\text{MDDH}_{k,n}^k$ assumption:

$$[\mathbf{A}]_1, [\mathbf{AS}]_1 \approx_c [\mathbf{A}]_1, [\mathbf{U}]_1$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$. This is analogous to "$\mathsf{G}_1 \approx_c \mathsf{G}_2$" for Π_1 in Section 4.3 except that we will use $\text{MDDH}_{k,n}^k$ instead of $\text{BI-MDDH}_{k,n}^k$ assumption. In particular, in the reduction, we simulate

$$\begin{aligned}
[\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [\mathbf{R}_i]_1 \ \text{ as } \ [\mathbf{R}_i]_2, [\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_1 \\
[\langle \mathbf{F}_i, \mathbf{AST}^\top \mathbf{B}\rangle - \tau_i]_2, [\tau_i]_1 \ \text{ as } \ [\tau_i]_2, [\langle \mathbf{F}_i, \mathbf{AST}^\top \mathbf{B}\rangle - \tau_i]_1
\end{aligned}$$

thanks to $\mathbf{R}_i \leftarrow \mathbb{Z}_p^{k\times k}$ and $\tau_i \leftarrow \mathbb{Z}_p$. This ensures that the reduction only requires $[\mathbf{A}]_1, [\mathbf{AS}]_1$ (on the right-hand side) instead of $[\mathbf{A}]_2, [\mathbf{AS}]_2$ (on the left-hand side) in order to simulate secret keys. See Lemma 6 for more details.

H_4**:** Identical to H_3 except that the challenge ciphertext and secret key for $\mathbf{F}_i$ are as follows:

$$\begin{aligned}\mathsf{ct}^* &= \big([(\mathbf{x}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\boxed{\mathbf{V}})\mathbf{M}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1), \widetilde{\mathsf{Enc}}_2(\widetilde{\mathsf{msk}}_2)\big)\\ \mathsf{sk}_{\mathbf{F}_i} &= \begin{pmatrix}\widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, ([\mathbf{A}^\top\mathbf{F}_i\mathbf{B}-\mathbf{R}_i]_2, [-\mathbf{r}_i]_2), [\langle\mathbf{F}_i, \mathbf{U}\boxed{\mathbf{V}^\top}\rangle - \tau_i]_2)\\ \widetilde{\mathsf{KeyGen}}_2(\widetilde{\mathsf{msk}}_2, ([\mathbf{R}_i]_1, [\mathbf{r}_i]_1), [\tau_i]_1)\end{pmatrix}\end{aligned}$$

where $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$ and $\mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$. We claim that $\mathsf{H}_3 \approx_c \mathsf{H}_4$. This follows from the $\mathrm{MDDH}_{k,m}^k$ assumption:

$$[\mathbf{B}]_2, [\mathbf{BT}]_2 \approx_c [\mathbf{B}]_2, [\mathbf{V}]_2$$

where $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$, $\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$. This is analogous to "$\mathsf{G}_2 \approx_c \mathsf{G}_3$" for Π_1 in Section 4.3. We omit the detail.

H_5**:** Identical to H_4 except that the challenge ciphertext and secret key for $\mathbf{F}$ are as follows:

$$\begin{aligned}\widetilde{\mathsf{ct}}^* &= \big(\boxed{[\widetilde{\mathbf{U}}]_1, [\widetilde{\mathbf{V}}]_2}, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1), \widetilde{\mathsf{Enc}}_2(\widetilde{\mathsf{msk}}_2)\big)\\ \widetilde{\mathsf{sk}}_{\mathbf{F}_i} &= \begin{pmatrix}\widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, ([\mathbf{A}^\top\mathbf{F}_i\mathbf{B}-\mathbf{R}_i]_2, [-\mathbf{r}_i]_2), \boxed{[\langle\mathbf{F}_i, \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top\rangle - \tau_i - \langle\mathbf{F}_i, \mathbf{xy}^\top\rangle]_2})\\ \widetilde{\mathsf{KeyGen}}_2(\widetilde{\mathsf{msk}}_2, ([\mathbf{R}_i]_1, [\mathbf{r}_i]_1), [\tau_i]_1)\end{pmatrix}\end{aligned}$$

where $\widetilde{\mathbf{U}} \leftarrow \mathbb{Z}_p^{n\times(k+1)}, \widetilde{\mathbf{V}} \leftarrow \mathbb{Z}_p^{m\times(k+1)}$. We claim that $\mathsf{H}_4 \approx_c \mathsf{H}_5$. This follows from the selective SIM-security of π_2 under $\mathsf{msk} = (\mathbf{U}, \mathbf{V})$. This is analogous to "$\mathsf{G}_3 \approx_c \mathsf{G}_4$" for Π_1 in Section 4.3. We omit the detail.

Note that H_5 can be simulated using the simulator described in Section 5.2 by setting $\mu = \langle\mathbf{F}, \mathbf{xy}^\top\rangle$.

5.4 Lemmas

Let $\mathsf{Adv}_i(\lambda)$ be the advantage function of $\mathcal{A}$ in G_i. We prove $\mathsf{H}_1 \approx_c \mathsf{H}_2$ and $\mathsf{H}_2 \approx_c \mathsf{H}_3$.

Lemma 5 ($\mathsf{H}_1 \approx_c \mathsf{H}_2$)**.** *There exists algorithm $\mathcal{B}_3$ such that* $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$|\mathsf{Adv}_1(\lambda) - \mathsf{Adv}_2(\lambda)| \le \mathsf{Adv}^{\text{BI-MDDH}_{d,Q}}_{\mathcal{B}_3}(\lambda).$$

Proof. This follows from the $\text{BI-MDDH}_{d,Q}$ assumption which implies that, for $\mathbf{S}, \mathbf{T}, \mathbf{R}_i \leftarrow \mathbb{Z}_p^{k\times k}$, we have

$$\begin{aligned}&\{[\mathbf{r}_i]_1, [\mathbf{r}_i]_2, [\langle\mathbf{R}_i, \mathbf{ST}^\top\rangle + \langle\mathbf{r}_i, \mathbf{s}\rangle]_1, [\langle\mathbf{R}_i, \mathbf{ST}^\top\rangle + \langle\mathbf{r}_i, \mathbf{s}\rangle]_2\}_{i\in[Q]}\\ \approx_c\ &\{[\mathbf{r}_i]_1, [\mathbf{r}_i]_2, [\tau_i]_1, \qquad [\tau_i]_2 \qquad\}_{i\in[Q]}\end{aligned}$$

where $\mathbf{r}_i, \mathbf{s} \leftarrow \mathbb{Z}_p^d$ and $\tau_i \leftarrow \mathbb{Z}_p$ for all $i \in [Q]$. On input

$$\mathbf{S}, \mathbf{T}, \{\mathbf{R}_i, [\mathbf{r}_i]_1, [\mathbf{r}_i]_2, [z_i]_1, [z_i]_2\}_{i\in[Q]}$$

where either $z_i = \langle\mathbf{R}_i, \mathbf{ST}^\top\rangle + \langle\mathbf{r}_i, \mathbf{s}\rangle$ or $z_i = \tau_i \leftarrow \mathbb{Z}_p$ for all $i \in [Q]$, the algorithm $\mathcal{B}_3$ works as follows:

(Setup) Run

$$(\mathsf{mpk}_1, \widetilde{\mathsf{msk}}_1) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k, 1^d),\ (\mathsf{mpk}_2, \widetilde{\mathsf{msk}}_2) \leftarrow \widetilde{\mathsf{Setup}}_2(1^\lambda, 1^k, 1^d)$$

and sample $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$. Output

$$\mathsf{mpk} = \big(\mathsf{mpk}_1, \mathsf{mpk}_2, [\mathbf{A}]_1, [\mathbf{B}]_2\big).$$

(Challenge) Once receiving the semi-adaptive challenge $(\mathbf{x}, \mathbf{y})$, create the challenge ciphertext as in H_1 (or H_2) using $\mathbf{A}, \mathbf{B}, \widetilde{\mathsf{msk}}_1, \widetilde{\mathsf{msk}}_2$ sampled during **Setup** and $\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ that are provided in the input.

(Key Queries) On input $\mathbf{F}_i$, sample $\mathbf{R}_i, \mathbf{r}_i$ honestly and output

$$\mathsf{sk}_{\mathbf{F}_i} = \begin{pmatrix} \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, ([\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [-\mathbf{r}_i]_2), [\langle \mathbf{F}_i, \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \rangle - z_i]_2) \\ \widetilde{\mathsf{KeyGen}}_2(\widetilde{\mathsf{msk}}_2, ([\mathbf{R}_i]_1, [\mathbf{r}_i]_1), [z_i]_1) \end{pmatrix}$$

using $[\mathbf{r}_i]_1, [\mathbf{r}_i]_2, [z_i]_1, [z_i]_2$ given out in the input.

(Analysis) Observe that, when $z_i = \langle \mathbf{R}_i, \mathbf{S}\mathbf{T}^\top \rangle + \langle \mathbf{r}_i, \mathbf{s} \rangle$, the simulation is identical to H_1; when $z_i = \tau_i \leftarrow \mathbb{Z}_p$, the simulation is identical to H_2. This readily proves the lemma. □

Lemma 6 ($\mathsf{H}_2 \approx_c \mathsf{H}_3$)**.** *There exists algorithm* $\mathcal{B}_4$ *such that* $\mathsf{Time}(\mathcal{B}_4) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$|\mathsf{Adv}_2(\lambda) - \mathsf{Adv}_3(\lambda)| \le \mathsf{Adv}^{\mathrm{MDDH}^k_{k,n}}_{\mathcal{B}_4}(\lambda).$$

Proof. This follows from $\mathrm{MDDH}^k_{k,n}$ assumption:

$$[\mathbf{A}]_1, [\mathbf{A}\mathbf{S}]_1 \approx_c [\mathbf{A}]_1, [\mathbf{U}]_1,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$. On input $[\mathbf{A}]_1$ and $[\mathbf{Z}]_1$ where either $\mathbf{Z} = \mathbf{A}\mathbf{S}$ or $\mathbf{Z} = \mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$, the algorithm $\mathcal{B}_4$ works as follows:

(Setup) Run

$$(\mathsf{mpk}_1, \widetilde{\mathsf{msk}}_1) \leftarrow \widetilde{\mathsf{Setup}}_1(1^\lambda, 1^k, 1^d), (\mathsf{mpk}_2, \widetilde{\mathsf{msk}}_2) \leftarrow \widetilde{\mathsf{Setup}}_2(1^\lambda, 1^k, 1^d)$$

and sample $\mathbf{B} \leftarrow \mathbb{Z}_p^{m\times k}$. Output

$$\mathsf{mpk} = \left(\mathsf{mpk}_1, \mathsf{mpk}_2, [\mathbf{A}]_1, [\mathbf{B}]_2\right)$$

using $[\mathbf{A}]_1$ given out in the input.

(Challenge) Once receiving the semi-adaptive challenge $(\mathbf{x}, \mathbf{y})$, sample $\mathbf{T} \leftarrow \mathbb{Z}_p^{k\times k}$ and $\mathbf{M}^*, \mathbf{M} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$ with $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Output

$$\mathsf{ct}^* = \left([(\mathbf{x}\|\mathbf{Z})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, \widetilde{\mathsf{Enc}}_1(\widetilde{\mathsf{msk}}_1), \widetilde{\mathsf{Enc}}_2(\widetilde{\mathsf{msk}}_2)\right)$$

using $[\mathbf{Z}]_1$ given out in the input.

(Key Queries) On input $\mathbf{F}_i$, sample $\mathbf{R}_i \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{r}_i \leftarrow \mathbb{Z}_p^d, \tau_i \leftarrow \mathbb{Z}_p$ and output

$$\mathsf{sk}_{\mathbf{F}_i} = \begin{pmatrix} \widetilde{\mathsf{KeyGen}}_1(\widetilde{\mathsf{msk}}_1, ([\mathbf{R}_i]_2, [-\mathbf{r}_i]_2), [\tau_i]_2) \\ \widetilde{\mathsf{KeyGen}}_2(\widetilde{\mathsf{msk}}_2, ([\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_1, [\mathbf{r}_i]_1), [\langle \mathbf{F}_i, \mathbf{Z}\mathbf{T}^\top \mathbf{B}^\top \rangle - \tau_i]_1) \end{pmatrix}$$

using $[\mathbf{A}]_1$ and $[\mathbf{Z}]_1$ given out in the input. Note that both $\mathbf{T}$ and $\mathbf{B}$ are known.

(Analysis) Observe that, when $\mathbf{Z} = \mathbf{A}\mathbf{S}$, the simulation is identical to H_2; especially, all keys given to the adversary has the same distribution as those in H_2 by the statistical argument

$$\overbrace{\begin{pmatrix} [\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [\mathbf{R}_i]_1, \\ [\langle \mathbf{F}_i, \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \rangle - \tau_i]_2, [\tau_i]_1 \end{pmatrix}_{i\in[Q]}}^{\mathsf{H}_2} \approx_s \overbrace{\begin{pmatrix} [\mathbf{R}_i]_2, [\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_1, \\ [\tau_i]_2, [\langle \mathbf{F}_i, \mathbf{A}\mathbf{S}\mathbf{T}^\top \mathbf{B}^\top \rangle - \tau_i]_1 \end{pmatrix}_{i\in[Q]}}^{\text{simulation with } \mathbf{Z}=\mathbf{A}\mathbf{S}};$$

where $\mathbf{R}_i \leftarrow \mathbb{Z}_p^{k\times k}$ and $\tau_i \leftarrow \mathbb{Z}_p$; when $\mathbf{Z} = \mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$, the simulation is identical to H_3; especially, all keys given to the adversary has the same distribution as those in H_3 by the statistical argument

$$\overbrace{\begin{pmatrix} [\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_2, [\mathbf{R}_i]_1, \\ [\langle \mathbf{F}_i, \mathbf{U}\mathbf{T}^\top \mathbf{B}^\top \rangle - \tau_i]_2, [\tau_i]_1 \end{pmatrix}_{i\in[Q]}}^{\mathsf{H}_3} \approx_s \overbrace{\begin{pmatrix} [\mathbf{R}_i]_2, [\mathbf{A}^\top \mathbf{F}_i \mathbf{B} - \mathbf{R}_i]_1, \\ [\tau_i]_2, [\langle \mathbf{F}_i, \mathbf{U}\mathbf{T}^\top \mathbf{B}^\top \rangle - \tau_i]_1 \end{pmatrix}_{i\in[Q]}}^{\text{simulation with } \mathbf{Z}=\mathbf{U}}.$$

where $\mathbf{R}_i \leftarrow \mathbb{Z}_p^{k\times k}$ and $\tau_i \leftarrow \mathbb{Z}_p$. This readily proves the lemma. □

6 Concrete Schemes

In this section, we present two concrete QFFE schemes instantiated from Π_1 and Π_2, respectively. For both of them, we use the parameter leading to the best efficiency. Recall that, we always instantiate (two-input) IPFE_1 and IPFE_2 using Wee's selectively SIM-secure construction under SXDH.

6.1 Concrete scheme from Bi-DLIN

We instantiate our first QFFE scheme Π_1 with $k = 2$. Namely, the semi-adaptive SIM-security is based on Bi-DLIN assumption as in [Gay20]. Our scheme has constant-size keys and shorter ciphertexts. The scheme is as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^m)$: Sample
$$\mathbf{A} \leftarrow \mathbb{Z}_p^{n\times 2}, \mathbf{B} \leftarrow \mathbb{Z}_p^{m\times 2}, \mathbf{d} \leftarrow \mathbb{Z}_p^2, \mathbf{W} \leftarrow \mathbb{Z}_p^{4\times 2}.$$
Output
$$\mathsf{mpk} = \left([\mathbf{A}]_1, [\mathbf{B}]_2, [\mathbf{d}, \mathbf{W}\mathbf{d}]_1\right) \quad \text{and} \quad \mathsf{msk} = \left(\mathbf{A}, \mathbf{B}, \mathbf{d}, \mathbf{W}\right).$$
- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}, \mathbf{y}))$: Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^m \times \mathbb{Z}_p^n$. Sample
$$(\mathbf{M}, \mathbf{M}^*) \leftarrow \mathbb{Z}_p^{3\times 3} \times \mathbb{Z}_p^{3\times 3}$$
such that $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Sample
$$\mathbf{S}, \mathbf{T} \leftarrow \mathbb{Z}_p^{2\times 2}, s \leftarrow \mathbb{Z}_p$$
and output
$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = \left([(\mathbf{x}\|\mathbf{A}\mathbf{S})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{B}\mathbf{T})\mathbf{M}]_2, [\mathsf{vec}(\mathbf{S}\mathbf{T}^\top) + \mathbf{W}\mathbf{d}s]_1, [\mathbf{d}s]_1\right) \in G_1^{3n} \times G_2^{3m} \times G_1^4 \times G_1^2.$$
- $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$: Let $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$. Output
$$\mathsf{sk}_\mathbf{F} = \left([\mathsf{vec}(\mathbf{A}^\top\mathbf{F}\mathbf{B})]_2, [\mathbf{W}^\top\mathsf{vec}(\mathbf{A}^\top\mathbf{F}\mathbf{B})]_2\right) \in G_2^4 \times G_2^2.$$
- $\mathsf{Dec}(\mathsf{ct}_{\mathbf{x},\mathbf{y}}, \mathsf{sk}_\mathbf{F})$: Parse
$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = ([\mathbf{C}_1]_1, [\mathbf{C}_2]_2, [\mathbf{c}_3]_1, [\mathbf{c}_4]_1) \quad \text{and} \quad \mathsf{sk}_\mathbf{F} = ([\mathbf{k}_1]_2, [\mathbf{k}_2]_2).$$
Compute
$$[L]_T \leftarrow e([\mathbf{c}_3^\top]_1, [\mathbf{k}_1]_2) \cdot e([\mathbf{c}_4^\top]_1, [\mathbf{k}_2]_2)^{-1},$$
$$[\mathbf{P}]_T = e([\mathbf{C}_1]_1, [\mathbf{C}_2^\top]_2), [D]_T = \langle \mathbf{F}, [\mathbf{P}]_T \rangle, [Z]_T = [D - L]_T$$
and recover Z from $[Z]_T$ via brute-force DLOG.

6.2 Concrete scheme from SXDH and Bi-DLIN

We instantiate our first QFFE scheme Π_2 with $k = 1$ and $d = 2$. Namely, the semi-adaptive SIM-security is based on SXDH and Bi-DLIN assumption. This gives even shorter keys and ciphertexts thanks to the smaller k. The scheme is as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^m)$: Sample
$$\mathbf{a} \leftarrow \mathbb{Z}_p^n, \mathbf{b} \leftarrow \mathbb{Z}_p^m, \mathbf{d}_1, \mathbf{d}_2 \leftarrow \mathbb{Z}_p^2, \mathbf{W}_1, \mathbf{W}_2 \leftarrow \mathbb{Z}_p^{3\times 2}.$$
Output
$$\mathsf{mpk} = \left([\mathbf{a}]_1, [\mathbf{b}]_2, [\mathbf{d}_1, \mathbf{W}_1\mathbf{d}_1]_1, [\mathbf{d}_2, \mathbf{W}_2\mathbf{d}_2]_2\right)$$
$$\mathsf{msk} = \left(\mathbf{a}, \mathbf{b}, \mathbf{d}_1, \mathbf{d}_2, \mathbf{W}_1, \mathbf{W}_2\right)$$

– $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}, \mathbf{y}))$: Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^m \times \mathbb{Z}_p^n$. Sample

$$(\mathbf{M}, \mathbf{M}^*) \leftarrow \mathbb{Z}_p^{2\times 2} \times \mathbb{Z}_p^{2\times 2}$$

such that $\mathbf{M}^*\mathbf{M}^\top = \mathbf{I}$. Sample

$$s, t, s_1, s_2 \leftarrow \mathbb{Z}_p, \mathbf{s} \leftarrow \mathbb{Z}_p^2$$

and output

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = \begin{pmatrix} [(\mathbf{x}\|\mathbf{a}s)\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{b}t)\mathbf{M}]_2 \\ [\binom{st}{\mathbf{s}} + \mathbf{W}_1\mathbf{d}_1 s_1]_1, [\mathbf{d}_1 s_1]_1 \\ [\binom{st}{\mathbf{s}} + \mathbf{W}_2\mathbf{d}_2 s_2]_2, [\mathbf{d}_2 s_2]_2 \end{pmatrix} \in G_1^{2n} \times G_2^{2m} \times G_1^3 \times G_1^2 \times G_2^3 \times G_2^2.$$

– $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{F})$: Let $\mathbf{F} \in \mathbb{Z}_p^{n\times m}$. Sample $r \leftarrow \mathbb{Z}_p$, $\mathbf{r} \leftarrow \mathbb{Z}_p^2$ and output

$$\mathsf{sk}_{\mathbf{F}} = \left(\left[\binom{\mathbf{a}^\top \mathbf{F}\mathbf{b} - r}{-\mathbf{r}} \right]_2, \left[\mathbf{W}_1^\top \cdot \binom{\mathbf{a}^\top \mathbf{F}\mathbf{b} - r}{-\mathbf{r}} \right]_2, [\binom{r}{\mathbf{r}}]_1, [\mathbf{W}_2^\top \binom{r}{\mathbf{r}}]_1 \right) \in G_2^3 \times G_2^2 \times G_1^3 \times G_1^2.$$

– $\mathsf{Dec}(\mathsf{ct}_{\mathbf{x},\mathbf{y}}, \mathsf{sk}_{\mathbf{F}})$: Parse

$$\mathsf{ct}_{\mathbf{x},\mathbf{y}} = ([\mathbf{C}_1]_1, [\mathbf{C}_2]_2, [\mathbf{c}_3]_1, [\mathbf{c}_4]_1, [\mathbf{c}_5]_2, [\mathbf{c}_6]_2), \mathsf{sk}_{\mathbf{F}} = ([\mathbf{k}_1]_2, [\mathbf{k}_2]_2, [\mathbf{k}_3]_1, [\mathbf{k}_4]_1)$$

Compute

$$[L_1]_T \leftarrow e([\mathbf{c}_3^\top]_1, [\mathbf{k}_1]_2) \cdot e([\mathbf{c}_4^\top]_1, [\mathbf{k}_2]_2)^{-1}, [L_2]_T \leftarrow e([\mathbf{k}_3^\top]_1, [\mathbf{c}_5]_2) \cdot e([\mathbf{k}_4^\top]_1, [\mathbf{c}_6]_2)^{-1},$$

$$[\mathbf{P}]_T = e([\mathbf{C}_1]_1, [\mathbf{C}_2^\top]_2), [D]_T = \langle \mathbf{F}, [\mathbf{P}]_T \rangle, [Z]_T = [D - L_1 - L_2]_T$$

and recover Z from $[Z]_T$ via brute-force DLOG.

Acknowledgement. We thank Romain Gay for helpful discussions and anonymous reviewers of ASIACRYPT 2020 for their useful comments and suggestions.

References

ABDP15. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 733–751. Springer, Heidelberg, March / April 2015.

AC17a. Shashank Agrawal and Melissa Chase. FAME: Fast attribute-based message encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 665–682. ACM Press, October / November 2017.

AC17b. Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 627–656. Springer, Heidelberg, April / May 2017.

ACF+18. Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 597–627. Springer, Heidelberg, August 2018.

AGRW17. Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 601–626. Springer, Heidelberg, April / May 2017.

AJ15. Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Heidelberg, August 2015.

ALMT20. Shweta Agrawal, Benoît Libert, Monosij Maitra, and Radu Titiu. Adaptive simulation security for inner product functional encryption. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 34–64. Springer, Heidelberg, May 2020.

ALS16. Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 333–362. Springer, Heidelberg, August 2016.

AS17. Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 152–181. Springer, Heidelberg, April / May 2017.

Att14. Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.

BCFG17. Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 67–98. Springer, Heidelberg, August 2017.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

BV15. Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015.

CDG+18. Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized multi-client functional encryption for inner product. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 703–732. Springer, Heidelberg, December 2018.

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015.

CGW18. Jie Chen, Junqing Gong, and Hoeteck Wee. Improved inner-product encryption with adaptive security and full attribute-hiding. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 673–702. Springer, Heidelberg, December 2018.

DGP18. Edouard Dufour Sans, Romain Gay, and David Pointcheval. Reading in the dark: Classifying encrypted digits with functional encryption. *IACR Cryptology ePrint Archive 2018/206*, 2018.

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

Gay20. Romain Gay. A new paradigm for public-key functional encryption for degree-2 polynomials. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 95–120. Springer, Heidelberg, May 2020.

GGH13a. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.

GGH+13b. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

GGHZ16. Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 480–511. Springer, Heidelberg, January 2016.

GJLS20. Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. *IACR Cryptology ePrint Archive 2020/764*, 2020.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

JLS19. Aayush Jain, Huijia Lin, and Amit Sahai. Simplifying constructions and assumptions for io. *IACR Cryptology ePrint Archive 2019/1252*, 2019.

Lin17. Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 599–629. Springer, Heidelberg, August 2017.

LL20. Huijia Lin and Ji Luo. Compact adaptively secure ABE from k-lin: Beyond NC^1 and towards NL. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 247–277. Springer, Heidelberg, May 2020.

O'N10. Adam O'Neill. Definitional issues in functional encryption. *IACR Cryptology ePrint Archive 2010/556*, 2010.

RPB+19. Theo Ryffel, David Pointcheval, Francis Bach, Edouard Dufour-Sans, and Romain Gay. Partially encrypted deep learning using functional encryption. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 4519–4530, 2019.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, August 2009.

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, February 2014.

Wee17. Hoeteck Wee. Attribute-hiding predicate encryption in bilinear groups, revisited. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 206–233. Springer, Heidelberg, November 2017.

A Selective SIM-Security of π_1 and π_2

In this section we sketch the proofs of selective SIM-security of π_1 and π_2, respectively. For this, we first sketch the proof for (5) shown in the Introduction in detail and describe the difference with π_1 and π_2, respectively.

Game Sequence for (5). To prove the selective SIM-security of (5), we will employ the following game sequence.

$\underline{\mathsf{G}_0}$**:** Real game.

$\underline{\mathsf{G}_{1.1}}$**:** Identical to G_0 except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ is:

$$[(\mathbf{x}+\boxed{\mathbf{Ua}}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V}-\boxed{\mathbf{ya}^\top})\mathbf{M}]_2$$

where $\mathbf{a} \leftarrow \mathbb{Z}_p^k$. We claim $\mathsf{G}_{1.1} \approx_s \mathsf{G}_0$. This follows from change of basis:

$$(\mathbf{M}^*, \mathbf{M}) \longmapsto (\mathbf{P}^*\mathbf{M}^*, \mathbf{PM}) \quad \text{where} \quad \mathbf{P}^* = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{a} & \mathbf{I} \end{pmatrix}, \mathbf{P} = \begin{pmatrix} 1 & -\mathbf{a}^\top \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

$\underline{\mathsf{G}_{1.2}}$**:** Identical to $\mathsf{G}_{1.1}$ except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ and a secret key for $\mathbf{F}$ are:

$$\begin{aligned} \mathsf{ct} &= [(\mathbf{x}+\mathbf{Ua}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V}-\cancel{\mathbf{ya}^\top})\mathbf{M}]_2; \\ \mathsf{sk}_{\mathbf{F}} &= [\langle \mathbf{F}, \mathbf{UV}^\top + \boxed{\mathbf{Uay}^\top} \rangle]_T \end{aligned}$$

where $\mathbf{a} \leftarrow \mathbb{Z}_p^k$. We claim $\mathsf{G}_{1.2} \approx_s \mathsf{G}_{1.1}$. This follows from change of variable:

$$\mathbf{V} \longmapsto \mathbf{V} + \mathbf{ya}^\top.$$

$\underline{\mathsf{G}_{1.3}}$**:** Identical to $\mathsf{G}_{1.2}$ except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ and a secret key for $\mathbf{F}$ are:

$$\begin{aligned}\mathsf{ct} &= [(\mathbf{x}+\boxed{\mathbf{u}}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}\|\mathbf{V})\mathbf{M}]_2;\\ \mathsf{sk}_{\mathbf{F}} &= [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top + \boxed{\mathbf{u}\mathbf{y}^\top}\rangle]_T\end{aligned}$$

where $\mathbf{u} \leftarrow \mathbb{Z}_p^n$. We claim $\mathsf{G}_{1.3} \approx_c \mathsf{G}_{1.2}$. This follows from the $\mathrm{MDDH}^1_{k,n}$ assumption:

$$[\mathbf{U}]_1, [\mathbf{U}\mathbf{a}]_1 \approx_c [\mathbf{U}]_1, [\mathbf{u}]_1$$

where $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times k}$, $\mathbf{a} \leftarrow \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow \mathbb{Z}_p^n$.

$\underline{\mathsf{G}_{2.1}}$**:** Identical to $\mathsf{G}_{1.3}$ except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ is:

$$[(\mathbf{x}+\mathbf{u}\|\mathbf{U}-\boxed{(\mathbf{x}+\mathbf{u})\mathbf{a}^\top})\mathbf{M}^*]_1, [(\mathbf{y}+\boxed{\mathbf{V}\mathbf{a}}\|\mathbf{V})\mathbf{M}]_2$$

where $\mathbf{a} \leftarrow \mathbb{Z}_p^k$. We claim $\mathsf{G}_{2.1} \approx_s \mathsf{G}_{1.3}$. This is analogous to $\mathsf{G}_{1.1} \approx_s \mathsf{G}_0$ and follows from change of basis:

$$(\mathbf{M}^*, \mathbf{M}) \longmapsto (\mathbf{P}\mathbf{M}^*, \mathbf{P}^*\mathbf{M})$$

where $\mathbf{P}, \mathbf{P}^*$ are defined as before. Here we exchange the role of $\mathbf{P}, \mathbf{P}^*$.

$\underline{\mathsf{G}_{2.2}}$**:** Identical to $\mathsf{G}_{2.1}$ except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ and a secret key for $\mathbf{F}$ are:

$$\begin{aligned}\mathsf{ct} &= [(\mathbf{x}+\mathbf{u}\|\mathbf{U}-\cancel{(\mathbf{x}+\mathbf{u})\mathbf{a}^\top})\mathbf{M}^*]_1, [(\mathbf{y}+\mathbf{V}\mathbf{a}\|\mathbf{V})\mathbf{M}]_2;\\ \mathsf{sk}_{\mathbf{F}} &= [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top + \boxed{(\mathbf{x}+\mathbf{u})\mathbf{a}^\top\mathbf{V}^\top} + \mathbf{u}\mathbf{y}^\top\rangle]_T\end{aligned}$$

where $\mathbf{a} \leftarrow \mathbb{Z}_p^k$. We claim $\mathsf{G}_{2.2} \approx_s \mathsf{G}_{2.1}$. This is analogous to $\mathsf{G}_{1.2} \approx_s \mathsf{G}_{1.1}$ and follows from change of variable:

$$\mathbf{U} \longmapsto \mathbf{U} + (\mathbf{x}+\mathbf{u})\mathbf{a}^\top.$$

$\underline{\mathsf{G}_{2.3}}$**:** Identical to $\mathsf{G}_{2.2}$ except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ and a secret key for $\mathbf{F}$ are:

$$\begin{aligned}\mathsf{ct} &= [(\mathbf{x}+\mathbf{u}\|\mathbf{U})\mathbf{M}^*]_1, [(\mathbf{y}+\boxed{\mathbf{v}}\|\mathbf{V})\mathbf{M}]_2;\\ \mathsf{sk}_{\mathbf{F}} &= [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top + \boxed{(\mathbf{x}+\mathbf{u})\mathbf{v}^\top} + \mathbf{u}\mathbf{y}^\top\rangle]_T\end{aligned}$$

where $\mathbf{v} \leftarrow \mathbb{Z}_p^m$. We claim $\mathsf{G}_{2.3} \approx_s \mathsf{G}_{2.2}$. This is analogous to $\mathsf{G}_{1.3} \approx_s \mathsf{G}_{1.2}$ and follows from the $\mathrm{MDDH}^1_{k,m}$ assumption:

$$[\mathbf{V}]_2, [\mathbf{V}\mathbf{a}]_2 \approx_c [\mathbf{V}]_2, [\mathbf{v}]_2$$

where $\mathbf{V} \leftarrow \mathbb{Z}_p^{m\times k}$, $\mathbf{a} \leftarrow \mathbb{Z}_p^k$ and $\mathbf{v} \leftarrow \mathbb{Z}_p^m$.

$\underline{\mathsf{G}_3}$**:** Identical to $\mathsf{G}_{2.3}$ except that the challenge ciphertext for $(\mathbf{x},\mathbf{y})$ and a secret key for $\mathbf{F}$ are:

$$\begin{aligned}\mathsf{ct} &= [(\boxed{\mathbf{u}}\|\mathbf{U})\mathbf{M}^*]_1, [(\boxed{\mathbf{v}}\|\mathbf{V})\mathbf{M}]_2;\\ \mathsf{sk}_{\mathbf{F}} &= [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top + \boxed{\mathbf{u}\mathbf{v}^\top - \mathbf{x}\mathbf{y}^\top}\rangle]_T\end{aligned}$$

where $\mathbf{u} \leftarrow \mathbb{Z}_p^n$ and $\mathbf{v} \leftarrow \mathbb{Z}_p^m$. We claim $\mathsf{G}_3 \approx_s \mathsf{G}_{2.3}$. This follows from change of variables:

$$(\mathbf{u}, \mathbf{v}) \longmapsto (\mathbf{u}-\mathbf{x}, \mathbf{v}-\mathbf{y}).$$

Finally, we note that the distribution in G_3 is identical to the simulator (see (4)) by setting

$$(\mathbf{u}\|\mathbf{U})\mathbf{M}^* = \widetilde{\mathbf{U}} \quad \text{and} \quad (\mathbf{v}\|\mathbf{V})\mathbf{M} = \widetilde{\mathbf{V}}$$

which are uniformly distributed over $\mathbb{Z}_p^{n\times(k+1)}$ and $\mathbb{Z}_p^{m\times(k+1)}$, respectively, and gives us $\mathbf{U}\mathbf{V}^\top + \mathbf{u}\mathbf{v}^\top = \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top$.

Game Sequence for π_1. The scheme π_1 is identical to (5) except that the secret key for $\mathbf{F}$ is over G_2:

$$\mathsf{sk}_{\mathbf{F}} = [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle]_2$$

The simulator and game sequence are also similar. The main difference is that we need $\text{BI-MDDH}^1_{k,n}$ to prove $\mathsf{G}_{1.3} \approx_c \mathsf{G}_{1.2}$ since $\mathbf{U}$ and $\mathbf{Ua}$ live over G_2 in $\mathsf{sk}_{\mathbf{F}}$.

Game Sequence for π_2. The scheme π_2 is identical to (5) except that the secret key for $\mathbf{F}$ consists of two elements from G_1 and G_2, respectively:

$$\mathsf{sk}_{\mathbf{F}} = \big([\tau]_1, [\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top \rangle - \tau]_2 \big).$$

The simulator and game sequence are also similar. The main difference is that when we prove $\mathsf{G}_{1.3} \approx_c \mathsf{G}_{1.2}$ we equivalently simulate the secret key as:

$$\mathsf{sk}_{\mathbf{F}} = \big([\langle \mathbf{F}, \mathbf{U}\mathbf{V}^\top + \boxed{\mathbf{U}\mathbf{a}\mathbf{y}^\top} \rangle - \tau]_1, [\tau]_2 \big)$$

such that the reduction only uses $[\mathbf{U}]_1, [\mathbf{Ua}]_1$ as for (5) and avoid the use of bilateral MDDH for π_1.