Multi-Input Inner-Product Functional Encryption from Pairings

Michel Abdalla[1,*], Romain Gay[1], Mariana Raykova[2,**], and Hoeteck Wee[1,***]

[1] ENS and PSL Research University, Paris
{michel.abdalla,romain.gay,hoeteck.wee}@ens.fr
[2] Yale University, New Haven
mariana.raykova@yale.edu

Abstract. We present a multi-input functional encryption scheme (MIFE) for the inner product functionality based on the k-Lin assumption in prime-order bilinear groups. Our construction works for any polynomial number of encryption slots and achieves adaptive security against unbounded collusion, while relying on standard polynomial hardness assumptions. Prior to this work, we did not even have a candidate for 3-slot MIFE for inner products in the generic bilinear group model. Our work is also the first MIFE scheme for a non-trivial functionality based on standard cryptographic assumptions, as well as the first to achieve polynomial security loss for a super-constant number of slots under falsifiable assumptions. Prior works required stronger non-standard assumptions such as indistinguishability obfuscation or multi-linear maps.

1   Introduction

In a functional encryption (FE) scheme [25, 11], an authority can generate restricted decryption keys that allow users to learn specific functions of the encrypted messages and nothing else. That is, each FE decryption key sk_f is associated with a function f and decrypting a ciphertext $\mathsf{Enc}(x)$ with sk_f results in $f(x)$. Multi-input functional encryption (MIFE) introduced by Goldwasser et al. [19] is a generalization of functional encryption to the setting of multi-input functions. A MIFE scheme has several encryption slots and each decryption key sk_f for a multi-input function f decrypts jointly ciphertexts $\mathsf{Enc}(x_1)$, ..., $\mathsf{Enc}(x_n)$ for all slots to obtain $f(x_1, \ldots, x_n)$ without revealing anything more about the encrypted messages. The MIFE functionality provides the capability to encrypt independently messages for different slots. This facilitates scenarios where information, which will be processed jointly during decryption, becomes available at different points of time or is provided by different parties. MIFE has many applications related to computation and data-mining over encrypted data coming from multiple sources, which include examples such as executing search queries over encrypted data, processing encrypted streaming data, non-interactive differentially private data releases, multi-client delegation of computation, order-revealing encryption [19, 10] (see Appendix B for applications). The security requirement for FE and MIFE is that the decryption keys are resilient to collusion attacks, namely any group of users holding different decryption keys learns nothing about the underlying messages beyond what each of them could individually learn.

We now have several constructions of MIFE schemes, which can be broadly classified as follows: (i) feasibility results for general circuits [19, 6, 5, 12], and (ii) constructions for specific functionalities, notably comparison, which corresponds to order-revealing encryption [10]. Unfortunately, all of these constructions rely on indistinguishability obfuscation, single-input FE for circuits, or multilinear maps [16, 15], which we do not know how to instantiate under standard and well-understood cryptographic assumptions.[3]

1.1   Our Contributions

In this work, we present a multi-input functional encryption scheme (MIFE) for the inner product functionality based on the k-Lin assumption in prime-order *bilinear* groups. This is the first MIFE scheme for a non-trivial functionality based on standard cryptographic assumptions with polynomial security loss, and for any polynomial number of slots and secure against unbounded collusions.

Concretely, the functionality we consider is that of "bounded-norm" multi-input inner product: each function is specified by a collection of n vectors $\mathbf{y}_1, \ldots, \mathbf{y}_n$, takes as input n vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and outputs

$$f_{\mathbf{y}_1, \ldots, \mathbf{y}_n}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{y}_i \rangle.$$

We require that the $\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}_n$ have bounded norm, and inner product is computed over the integers. The functionality is a natural generalization of single-input inner product functionality introduced by Abdalla et. al [1], and studied in [1, 7, 13, 4, 2], and captures several useful computations arising in the context of data-mining. A summary of our results and prior works on single-input inner product is shown in Fig. 1.

---

[3] In this paper, we refer only to unbounded collusions (i.e. the adversary can request for any number of secret keys). See [24, 21, 20, 12] for results on bounded collusions.

Prior approaches. Prior constructions of MIFE schemes in [10] require (at least) nm-linear maps for n slots with m-bit inputs as they encode each input bit for each slot into a fresh level of a multi-linear map. In addition, there is typically a security loss that is exponential in n due to the combinatorial explosion arising from combining different ciphertexts across the slots. In the case of inner product, one can hope to reduce the multi-linearity to n by exploiting linearity as in the single-input FE; indeed, this was achieved in two independent works [23, 22][4] showing how to realize a two-slot MIFE for inner product over bilinear groups. We stress that our result is substantially stronger: we show how to realize n-slot MIFE for inner product for any polynomial n over bilinear groups under standard assumptions, while in addition avoiding the exponential security loss. In particular, we deviate from the prior approaches of encoding each slot into a fresh level of a multi-linear map. We stress that prior to this work, we do not even have a candidate for 3-slot MIFE for inner product in the generic bilinear group model.

A public-key scheme. Our first observation is that we can build a public-key MIFE for inner product by running n independent copies of a single-input FE for inner product. Combined with existing instantiations of the latter in [1], this immediately yields a public-key MIFE for inner product under the standard DDH in cyclic groups.

In a bit more detail, we recall the DDH-based public-key single-input FE scheme from [1]:[5]

$$\mathsf{mpk} := [\mathbf{w}], \ \mathsf{ct}_{\mathbf{x}} = ([s], [\mathbf{x} + \mathbf{w}s]), \ \mathsf{sk}_{\mathbf{y}} := \langle \mathbf{w}, \mathbf{y} \rangle.$$

Decryption computes $[\langle \mathbf{x}, \mathbf{y} \rangle] = [\mathbf{x} + \mathbf{w}s]^\top \mathbf{y} \cdot [s]^{-\langle \mathbf{w}, \mathbf{y} \rangle}$ and then recovers $\langle \mathbf{x}, \mathbf{y} \rangle$ by computing the discrete log.

Our public-key MIFE scheme is as follows:

$$\mathsf{mpk} := ([\mathbf{w}_1], \ldots, [\mathbf{w}_n]),$$
$$\mathsf{ct}_{\mathbf{x}_i} := ([s_i], [\mathbf{x}_i + \mathbf{w}_i s_i]),$$
$$\mathsf{sk}_{\mathbf{y}_1, \ldots, \mathbf{y}_n} := (\langle \mathbf{w}_1, \mathbf{y}_1 \rangle, \ldots, \langle \mathbf{w}_n, \mathbf{y}_n \rangle).$$

We note that the encryption of $\mathbf{x}_i$ uses fresh randomness s_i; to decrypt, we need to know each $\langle \mathbf{w}_i, \mathbf{y}_i \rangle$, and not just $\langle \mathbf{w}_1, \mathbf{y}_1 \rangle + \cdots + \langle \mathbf{w}_n, \mathbf{y}_n \rangle$. In particular, an adversary can easily recover each $[\langle \mathbf{x}_i, \mathbf{y}_i \rangle]$, whereas the ideal functionality should only leak the sum $\sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$. In the *public-key* setting, it is easy to see that $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$ is in fact inherent leakage from the ideal functionality. Concretely, an adversary can always pad an encryption of $\mathbf{x}_i$ in the i'th slot with encryptions of $\mathbf{0}$'s in the remaining $n-1$ slots and then decrypt.

Our main scheme. The bulk of this work lies in constructing a multi-input FE for inner product in the *private-key* setting, where we can no longer afford to leak $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$. We modify the previous scheme by introducing additional rerandomization into each slot with the use of bilinear groups as follows:

$$\mathsf{msk} := ([\mathbf{w}_1]_1, [v_1]_1, [z_1]_1, \ldots, [\mathbf{w}_n]_1, [v_n]_1, [z_n]_1),$$
$$\mathsf{ct}_{\mathbf{x}_i} := ([s_i]_1, [\mathbf{x}_i + \mathbf{w}_i s_i]_1, [z_i + v_i s_i]_1),$$
$$\mathsf{sk}_{\mathbf{y}_1, \ldots, \mathbf{y}_n} := ([\langle \mathbf{w}_1, \mathbf{y}_1 \rangle + v_1 r]_2, \ldots, [\langle \mathbf{w}_n, \mathbf{y}_n \rangle + v_n r]_2, [r]_2, [(z_1 + \cdots + z_n)r]_T).$$

---

[4] This work is independent of both works.

[5] Here, we use the implicit representation notation for group elements, using $[s]$ to denote g^s and $[\mathbf{w}]$ to denote $g^{\mathbf{w}}$, etc.

The ciphertext $\mathsf{ct}_{\mathbf{x}_i}$ can be viewed as encrypting $\mathbf{x}_i \| z_i$ using the single-input FE, where $z_1, \ldots, z_n$ are part of msk. In addition, we provide a single-input FE key for $\mathbf{y}_i \| r$ in the secret key, where a fresh r is sampled for each key. Decryption proceeds as follows: first compute

$$[\langle \mathbf{x}_i, \mathbf{y}_i \rangle + z_i r]_T = e([\mathbf{x}_i + \mathbf{w}_i s_i]_1^\top, [\mathbf{y}_i]_2) \cdot e([z_i + v_i s_i]_1^\top, [r]_2) \cdot e([s_i], [\langle \mathbf{w}_i, \mathbf{y}_i \rangle + v_i r]_2)^{-1}$$

and then

$$[\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{y}_i \rangle]_T = [(z_1 + \cdots + z_n)r]_T^{-1} \cdot \prod_{i=1}^{n} [\langle \mathbf{x}_i, \mathbf{y}_i \rangle + z_i r]_T.$$

The intuition underlying security is that by the DDH assumption $[z_i r]_T$ is pseudorandom and helps mask the leakage about $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$ in $[\langle \mathbf{x}_i, \mathbf{y}_i \rangle + z_i r]_T$; in particular,

$$[\langle \mathbf{x}_1, \mathbf{y}_1 \rangle + z_1 r]_T, \ldots, [\langle \mathbf{x}_n, \mathbf{y}_n \rangle + z_n r]_T, [(z_1 + \cdots + z_n)r]_T$$

constitutes a computational secret-sharing of $[\langle \mathbf{x}_1, \mathbf{y}_1 \rangle + \cdots + \langle \mathbf{x}_n, \mathbf{y}_n \rangle]_T$, even upon reusing $z_1, \ldots, z_n$ as long as we pick a fresh r. In addition, sharing the same exponent r across n elements in the secret key helps prevent mix-and-match attacks across secret keys.

Our main technical result is that a variant of the private-key MIFE scheme we just described satisfies adaptive indistinguishability-based security under the k-Lin assumption in bilinear groups; a straight-forward extension of an impossibility result in [11, 3] rules out simulation-based security. Our final scheme, described in Fig. 10, remains quite simple and achieves good concrete efficiency. We focus on selective security in this overview, and explain at the end the additional ideas needed to achieve adaptive security.

Overview of security proof. There are two main challenges in the security proof: (i) avoiding leakage beyond the ideal functionality, (ii) avoiding super-polynomial hardness assumptions. Our proof proceeds in two steps: first, we establish security with a single challenge ciphertext per slot, and from which we bootstrap to achieve security with multiple challenge ciphertexts per slot. We will address the first challenge in the first step and the second challenge in the second. For notation simplicity, we focus on the setting with $n = 2$ slots and a single key query $\mathbf{y}_1 \| \mathbf{y}_2$.

Step 1. To prove indistinguishability-based security, we want to switch encryptions $\mathbf{x}_1^0, \mathbf{x}_2^0$ to encryptions of $\mathbf{x}_1^1, \mathbf{x}_2^1$. Here, the leakage from the ideal functionality imposes the restriction that

$$\langle \mathbf{x}_1^0, \mathbf{y}_1 \rangle + \langle \mathbf{x}_2^0, \mathbf{y}_2 \rangle = \langle \mathbf{x}_1^1, \mathbf{y}_1 \rangle + \langle \mathbf{x}_2^1, \mathbf{y}_2 \rangle$$

and this is the only restriction we can work with. The natural proof strategy is to introduce an intermediate hybrid that generates encryptions of $\mathbf{x}_1^1, \mathbf{x}_2^0$. However, to move from encryptions $\mathbf{x}_1^0, \mathbf{x}_2^0$ to this hybrid, we would require that $\langle \mathbf{x}_1^0 \| \mathbf{x}_2^0, \mathbf{y}_1 \| \mathbf{y}_2 \rangle = \langle \mathbf{x}_1^1 \| \mathbf{x}_2^0, \mathbf{y}_1 \| \mathbf{y}_2 \rangle$, which implies the extraneous restriction $\langle \mathbf{x}_1^0, \mathbf{y}_1 \rangle = \langle \mathbf{x}_1^1, \mathbf{y}_1 \rangle$. (Indeed, the single-input inner product scheme in [7] imposes extraneous restrictions to overcome similar difficulties in the function-hiding setting.)

To overcome this challenge, we rely on a single-input FE that achieves simulation-based security, which allows us to avoid the intermediate hybrid. See Theorem 1 and Remark 4 for further details.

Step 2. Next, we consider the more general setting with Q_1 challenge ciphertexts in the first slot and Q_2 in the second, but still a single key query. We achieve security loss $O(Q_1 + Q_2)$ for two slots, and more generally, $O(Q_1 + \cdots + Q_n)$ —as opposed to $Q_1 Q_2 \cdots Q_n$ corresponding to all possible combinations of the challenge ciphertexts— for n slots.

Reference	# inputs	setting	security	assumption	pairing
ABDP15 [1]	1	public-key	many-SEL-IND	DDH	no
ALS15 [4], ABDP16 [2]	1	public-key	many-AD-IND	DDH, k-Lin	no
BSW11 [11]	1	any	many-SEL-SIM	impossible	
[28]	1	public-key	one-SEL-SIM	k-Lin	no
LL16 [23]	2	private-key	many-SEL-IND	SXDH + T3DH	yes
KLMMRW16 [22]	2	private-key	single-key many-AD-IND[6]	function-private FE	yes
easy	multi	public-key	many-AD-IND	k-Lin	no
this work	multi	private-key	many-AD-IND	k-Lin	yes

Fig. 1: Summary of constructions from cyclic or bilinear groups. We have 8 security notions xx-yy-zzz where xx $\in$ {one, many} refers to the number of challenge ciphertexts; yy $\in$ {SEL, AD} refers to the fact that encryption queries are selectively or adaptively chosen; zzz $\in$ {IND, SIM} refers to indistinguishability vs simulation-based security.

Our first observation is that we can bound the leakage from the ideal functionality by $O(Q_1 + Q_2)$ relations (the trivial bound being $Q_1 \cdot Q_2$). Denote the j'th ciphertext query in the i'th slot by $\mathbf{x}_i^{j,b}$, where b is the challenge bit. By decrypting the encryptions of $\mathbf{x}_1^{2,b}, \mathbf{x}_2^{1,b}$ and $\mathbf{x}_1^{1,b}, \mathbf{x}_2^{1,b}$ and substracting the two, the adversary learns $\langle \mathbf{x}_1^{2,b} - \mathbf{x}_1^{1,b}, \mathbf{y}_1 \rangle$ and more generally, $\langle \mathbf{x}_i^{j,b} - \mathbf{x}_i^{1,b}, \mathbf{y}_i \rangle$. Indeed, these are essentially the only constraints we need to work with, namely:

$$\langle \mathbf{x}_1^{1,0}, \mathbf{y}_1 \rangle + \langle \mathbf{x}_2^{1,0}, \mathbf{y}_2 \rangle = \langle \mathbf{x}_1^{1,1}, \mathbf{y}_1 \rangle + \langle \mathbf{x}_2^{1,1}, \mathbf{y}_2 \rangle,$$
$$\langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle, j = 2, \ldots, Q_i, i = 1, 2.$$

Next, we need to translate the bound on the constraints to a $O(Q_1 + Q_2)$ bound on the security loss in the security reduction. We will switch from encryptions of $\mathbf{x}_i^{j,0}$ to those of $\mathbf{x}_i^{j,1}$ as follows: we write

$$\mathbf{x}_i^{j,0} = \mathbf{x}_i^{1,0} + (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}).$$

We can switch the first terms in the sums from $\mathbf{x}_i^{1,0}$ to $\mathbf{x}_i^{1,1}$ using security for a single challenge ciphertext, and then switch $\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}$ to $\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}$ by relying on security of the underlying single-input FE and the fact that $\langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle$. Here, we will require that the underlying single-input FE satisfies a malleability property, namely given Δ, we can maul an encryption of $\mathbf{x}$ into that of $\mathbf{x} + \Delta$. Note that this does not violate security because given $\langle \mathbf{x}, \mathbf{y} \rangle, \mathbf{y}, \Delta$, we can efficiently compute $\langle \mathbf{x} + \Delta, \mathbf{y} \rangle$. See Theorem 2 for further details.

Extension to adaptive security. The previous argument for selective security requires to embed the challenge into the setup parameters. To circumvent this issue, we use a two-step strategy for the adaptive security proof of MIFE. The first step uses an adaptive argument (this is essentially the

---

[6] The security notion achieved in [22] is actually a weaker variant of many-AD-IND in which the adversary is only allowed to perform a single key query at the beginning of the security game.

argument used for the selective case, but applied to parameters that are picked at setup time), while the second step uses a selective argument, with *perfect security*. Thus, we can afford using complexity leveraging, which incurs an exponential security loss, since the exponential term is multiplied by a zero term. The idea of using complexity leveraging to deduce adaptive security from selective security when the security is perfect, already appears in [27, Remark 1]. See Remark 5 for further details.

Theoretical perspective. The focus of this work is on obtaining constructions for a specific class of functions with good concrete efficiency. Nonetheless, we believe that our results do shed some new insights into general feasibility results for MIFE:

– First, our results are indicative of further qualitative differences between MIFE in the public-key and the private-key settings. Indeed, we already know that the security guarantees are quite different due to additional inherent leakages in the public-key setting. In the case of order-revealing encryption [10], the differences are sufficient to enable positive results in the private-key setting, while completely ruling out any construction in the public-key setting. Our results hint at a different distinction, where the private-key setting seems to require qualitative stronger assumptions than in the public-key setting, namely the use of pairings.

– Next, our results provide the first evidence supporting the intuition that MIFE requires qualitatively stronger assumptions than FE, but not too much stronger. Concretely, for the inner product FE, we have existing positive results under the DDH assumption in pairing-free groups. Prior to this work, it was not clear if we could extend the positive results to MIFE for n-ary inner product under the same assumptions, or if n-ary inner product would already require the same complex assumptions as MIFE for circuits. Our results suggest a rather different picture, namely that going from single-input to multi-input should require no more than an extra level of multi-linearity, even for restricted functionalities. The situation is somewhat different for general circuits, where we now know that going from single-input to multi-input incurs no more than a quantitative loss in the underlying assumptions [5, 12].

– Finally, we presented the first MIFE for a non-trivial functionality that polynomial security loss for a super-constant number of slots under falsifiable assumptions. Recall that indistinguishability obfuscation and generic multi-linear maps are not falsifiable, whereas the constructions based on single-input FE in [5, 8, 12] incur a security loss which is exponential in the number of slots. Indeed, there is a reason why prior works relied on non-falsifiable assumptions or super-polynomial security loss. Suppose an adversary makes Q_0 key queries, and $Q_1, \ldots, Q_n$ ciphertext queries for the n slots. By combining the ciphertexts and keys in different ways, the adversary can learn $Q_0 Q_1 \cdots Q_n$ different decryptions. When n is super-constant, the winning condition in the security game may not be efficiently checkable in polynomial-time, hence the need for either a non-falsifiable assumption or a super-polynomial security loss. To overcome this difficulty, we show that for inner product, we can exploit linearity to succinctly characterize the $Q_0 Q_1 \cdots Q_n$ constraints by roughly $Q_0 \cdot (Q_1 + \cdots Q_n)$ constraints.

1.2 Discussion

Beyond inner product? Our constructions and techniques may seem a-priori largely tailored to the inner product functionality and properties of bilinear groups. We clarify here that our high-level approach (which builds upon [27, 9]) may be applicable beyond inner product, namely:

i. start with a multi-input FE that is only secure for a single ciphertext per slot and one secret key, building upon a single-input FE whose security is simulation-based for a single ciphertext (in our case, this corresponds to introducing the additional $z_1, \ldots, z_n$ to hide the intermediate computation $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$);

ii. achieve security for a single ciphertext per slot and multiple secret keys, by injecting additional randomness to the secret keys to prevent mix-and-match attacks (for this, we replaced $z_1, \ldots, z_n$ with $z_1 r, \ldots, z_n r, r$ in the exponent);

iii. "bootstrap" to multiple ciphertexts per slot, where we also showed how to avoid incurring an exponential security loss.

In particular, using simulation-based security for i. helped us avoid additional leakage beyond what is allowed by the ideal functionality.

Additional related work. Goldwasser et al. [19] showed that both two-input public-key MIFE as well as n-input private-key MIFE for circuits already implies indistinguishability obfuscation for circuits.

There have also been several works that proposed constructions for private-key multi-input functional encryption. The work of Boneh et al. [10] constructs a single-key MIFE in the private key setting, which is based on multilinear maps and is proven secure in the idealized generic multilinear map model. Two other papers explore the question how to construct multi-input functional encryption starting from the single input variant. In their work [5] Ananth and Jain demonstrate how to obtain selectively secure MIFE in the private key setting starting from any general-purpose public key functional encryption. In an independent work, Brakerski et al. [12] reduce the construction of private key MIFE to general-purpose private key (single input) functional encryption. The resulting scheme achieves selective security when the starting private key FE is selectively secure. Additionally in the case when the MIFE takes any constant number of inputs, adaptive security for the private key FE suffices to obtain adaptive security for the MIFE construction as well. The constructions in that work provide also function hiding properties for the MIFE encryption scheme.

While this line of work reduces MIFE to single-input FE for general-purpose constructions, the only known instantiations of construction for public and private key functional encryption with unbounded number of keys require either indistinguishability obfuscation [16] or multilinear maps with non-standard assumptions [17]. We stress that the transformations from single-input to MIFE in [5, 12] are not applicable in the case of inner product since these transformations require that the single-input FE for complex functionalities related to computing a PRF, which is not captured by the simple inner functionality.

Open problems. One natural open problem is to eliminate the use of pairings in MIFE for inner product; we think such a result would be quite surprising though. Another open problem is to achieve function privacy, as considered in the setting of single-input inner product functional encryption in [7, 13]. Note that these latter results require pairings. Our first guess is that it would be possible to achieve private-key, function-hiding MIFE for inner product under the k-Lin assumption in bilinear groups.

2 Preliminaries

Notation. We denote by $s \leftarrow_R S$ the fact that s is picked uniformly at random from a finite set S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout, we use 1^λ as the security parameter. We use lower case boldface to denote (column) vectors and upper case boldface to denote matrices.

Cryptographic assumptions We follow the notation and algebraic framework for Diffie-Hellman-like assumptions in [14]. We fix a pairing group $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ of prime order q, where q is a prime of $\Theta(\lambda)$ bits. We use the implicit representation notation for group elements: for fixed generators g_1 and g_2 of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and for a matrix $\mathbf{M}$ over $\mathbb{Z}_q$, we define $[\mathbf{M}]_1 := g_1^{\mathbf{M}}$ and $[\mathbf{M}]_2 := g_2^{\mathbf{M}}$, where exponentiation is carried out component-wise.

We recall the definitions of the Matrix Decision Diffie-Hellman (MDDH) Assumption [14].

Definition 1 (Matrix Distribution). *Let $k, \ell \in \mathbb{N}$, with $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs matrices in $\mathbb{Z}_q^{\ell \times k}$ of full rank k in polynomial time. We write $\mathcal{D}_k := \mathcal{D}_{k+1,k}$.*

Without loss of generality, we assume the first k rows of $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$ form an invertible matrix. The $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman problem is to distinguish the two distributions $([\mathbf{A}], [\mathbf{Aw}])$ and $([\mathbf{A}], [\mathbf{u}])$ where $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$, $\mathbf{w} \leftarrow_R \mathbb{Z}_q^k$ and $\mathbf{u} \leftarrow_R \mathbb{Z}_q^\ell$.

Definition 2 ($\mathcal{D}_k$-Matrix Diffie-Hellman Assumption $\mathcal{D}_k$-MDDH). *Let $\mathcal{D}_k$ be a matrix distribution. We say that the $\mathcal{D}_k$-Matrix Diffie-Hellman ($\mathcal{D}_k$-MDDH) Assumption holds relative to $\mathcal{PG}$ in $\mathbb{G}_s$ for $s \in \{1, 2\}$, if for all PPT adversaries $\mathcal{A}$, there exists a negligible function Adv such that:*

$$\mathsf{Adv}_{\mathbb{G}_s, \mathcal{A}}^{\mathcal{D}_k\text{-MDDH}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{Aw}]_s) = 1] - \Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{u}]_s) = 1]| = \mathsf{negl}(\lambda),$$

where the probability is taken over $\mathbf{A} \leftarrow_R \mathcal{D}_k, \mathbf{w} \leftarrow_R \mathbb{Z}_q^k, \mathbf{u} \leftarrow_R \mathbb{Z}_q^{k+1}$.

For each $k \geq 1$, [14] specifies distributions $\mathcal{L}_k$, $\mathcal{SC}_k$, $\mathcal{C}_k$ (and others) over $\mathbb{Z}_q^{(k+1) \times k}$ such that the corresponding $\mathcal{D}_k$-MDDH assumptions are generically secure in bilinear groups and form a hierarchy of increasingly weaker assumptions. $\mathcal{L}_k$-MDDH is the well known k-Linear Assumption k-Lin with 1-Lin = DDH. In this work we are mostly interested in the uniform matrix distribution $\mathcal{U}_{\ell,k}$.

Definition 3 (Uniform distribution). *Let $\ell, k \in \mathbb{N}$, with $\ell > k$. We denote by $\mathcal{U}_{\ell,k}$ the uniform distribution over all full-rank $\ell \times k$ matrices over $\mathbb{Z}_q$. Let $\mathcal{U}_k := \mathcal{U}_{k+1,k}$.*

Let $Q \geq 1$. For $\mathbf{W} \leftarrow_R \mathbb{Z}_q^{k \times Q}, \mathbf{U} \leftarrow_R \mathbb{Z}_q^{(k+1) \times Q}$, we consider the Q-fold $\mathcal{U}_{\ell,k}$-MDDH Assumption which consists in distinguishing the distributions $([\mathbf{A}], [\mathbf{AW}])$ from $([\mathbf{A}], [\mathbf{U}])$. That is, a challenge for the Q-fold $\mathcal{U}_{\ell,k}$-MDDH Assumption consists of Q independent challenges of the $\mathcal{U}_{\ell,k}$-MDDH Assumption (with the same $\mathbf{A}$ but different randomness $\mathbf{w}$). We recall in Lemma 1 the random self reducibility of the Q-fold $\mathcal{U}_{\ell,k}$-MDDH assumption, namely, the fact that it reduces to the 1-fold $\mathcal{U}_k$ assumption.

Lemma 1 ($\mathcal{U}_k$-MDDH $\Rightarrow$ Q-fold $\mathcal{U}_{\ell,k}$-MDDH [14, 18]). *Let $\ell, k \in \mathbb{N}^*$, with $\ell > k$, and $s \in \{1, 2\}$. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}_{\mathbb{G}_s,\mathcal{A}}^{Q\text{-}\mathcal{U}_{\ell,k}\text{-MDDH}}(\lambda) \leq \mathsf{Adv}_{\mathbb{G}_s,\mathcal{B}}^{\mathcal{U}_k\text{-MDDH}}(\lambda) + \frac{1}{q-1},$$

where $\mathsf{Adv}_{\mathbb{G}_s,\mathcal{A}}^{Q\text{-}\mathcal{U}_{\ell,k}\text{-MDDH}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{AW}]_s) = 1] - \Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}], [\mathbf{U}]) = 1]|$ and the probability is taken over $\mathbf{A} \leftarrow_\mathrm{R} \mathcal{U}_{\ell,k}, \mathbf{W} \leftarrow_\mathrm{R} \mathbb{Z}_q^{k \times Q}, \mathbf{U} \leftarrow_\mathrm{R} \mathbb{Z}_q^{(k+1) \times Q}$.

Among all possible matrix distributions $\mathcal{D}_k$, the uniform matrix distribution $\mathcal{U}_k$ is the hardest possible instance, so in particular k-Lin $\Rightarrow \mathcal{U}_k$-MDDH, as stated in Lemma 2.

Lemma 2 ($\mathcal{D}_k$-MDDH $\Rightarrow \mathcal{U}_k$-MDDH, [14]). *Let $\mathcal{D}_k$ be a matrix distribution. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that $\mathsf{Adv}_{\mathbb{G}_s,\mathcal{B}}^{\mathcal{U}_k\text{-MDDH}}(\lambda) \leq \mathsf{Adv}_{\mathbb{G}_s,\mathcal{A}}^{\mathcal{D}_k\text{-MDDH}}(\lambda)$.*

3 Definitions for Multi-Input Functional Encryption

We recall the definitions for multi-input functional encryption from [19]. We focus here on the private-key setting, which allows us to simplify the definitions.

Definition 4 (Multi-input Function Encryption). *Let $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ be an ensemble where each $\mathcal{F}_n$ is a family of n-ary functions. A function $f \in \mathcal{F}_n$ is defined as follows $f : \mathcal{X}_1 \times \ldots \times \mathcal{X}_n \to \mathcal{Y}$. A multi-input functional encryption scheme $\mathcal{MIFE}$ for $\mathcal{F}$ consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda, \mathcal{F}_n)$: *on input the security parameter λ and a description of $\mathcal{F}_n \in \mathcal{F}$, outputs a master public key mpk[7] and a master secret key msk. All of the remaining algorithms get mpk as part of its input.*
- $\mathsf{Enc}(\mathsf{msk}, i, x_i)$: *on input the master secret key msk, $i \in [n]$, and a message $x_i \in \mathcal{X}_i$, outputs a ciphertext ct. We assume that each ciphertext has an associated index i, which denotes what slot this ciphertext can be used for. If $n = 1$, we omit the input i.*
- $\mathsf{KeyGen}(\mathsf{msk}, f)$: *on input the master secret key msk and a function $f \in \mathcal{F}_n$, outputs a decryption key sk_f.*
- $\mathsf{Dec}(\mathsf{sk}_f, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$: *on input a decryption key sk_f for function f and n ciphertexts, outputs a string $y \in \mathcal{Y}$.*

The scheme $\mathcal{MIFE}$ is correct if for all $f \in \mathcal{F}$ and all $x_i \in \mathcal{X}_i$ for $1 \leq i \leq n$, we have

$$\Pr\left[\begin{array}{r} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, n); \\ \mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f); \\ \mathsf{Dec}(\mathsf{sk}_f, f, \mathsf{Enc}(\mathsf{msk}, 1, x_1), \ldots, \mathsf{Enc}(\mathsf{msk}, n, x_n)) = f(x_1, \ldots, x_n) \end{array}\right] = 1,$$

where the probability is taken over the coins of Setup, KeyGen and Enc.

---

[7] We note that in the private key setting of MIFE, we can make mpk part of msk, but we allow for a separate master public key for better clarity in our proofs. In constructions where we do not need mpk we omit it.

3.1 Security notions

Following [3], we may consider 8 security notions xx-yy-zzz where xx $\in$ {one, many} refers to the number of challenge ciphertexts; yy $\in$ {SEL, AD} refers to the fact that encryption queries are selectively or adaptively chosen; zzz $\in$ {IND, SIM} refers to indistinguishability vs simulation-based security. We have the following trivial relations: many $\Rightarrow$ one, AD $\Rightarrow$ SEL, and the following standard relations: SIM $\Rightarrow$ IND, and one-yy-IND $\Rightarrow$ many-yy-IND, the latter in the public-key setting. Here, we focus on {one,many}-SEL-IND and one-SEL-SIM. We recall the definitions for adaptive security in Section 5.

Definition 5 (xx-SEL-IND-secure MIFE). *For every multi-input functional encryption* $\mathcal{MIFE} := (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *for* $\mathcal{F}$*, every security parameter* λ*, every stateful adversary* $\mathcal{A}$*, and every* $xx \in \{$one,many$\}$*, the advantage of* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}^{\mathcal{MIFE}, xx\text{-}SEL\text{-}IND}(\lambda, \mathcal{A})$$
$$= \left| \Pr\left[\mathbf{xx\text{-}SEL\text{-}IND}_0^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) = 1 \right] - \Pr\left[\mathbf{xx\text{-}SEL\text{-}IND}_1^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) = 1 \right] \right|$$
$$= \left| 1 - 2 \cdot \Pr\left[\mathbf{xx\text{-}SEL\text{-}IND}^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) = 1 \right] \right|$$

where the experiments are defined as follows:

$\underline{Experiment}\ \mathbf{xx\text{-}SEL\text{-}IND}_\beta^{\mathcal{MIFE}}(1^\lambda, \mathcal{A})$:	$\underline{Experiment}\ \mathbf{xx\text{-}SEL\text{-}IND}^{\mathcal{MIFE}}(1^\lambda, \mathcal{A})$:
	$\beta \leftarrow_{\mathrm{R}} \{0,1\}$
$\{x_i^{j,b}\}_{i\in[n], j\in[Q_i], b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n)$	$\{x_i^{j,b}\}_{i\in[n], j\in[Q_i], b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n)$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_n)$	$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_n)$
$\mathsf{ct}_i^j \leftarrow \mathsf{Enc}(\mathsf{msk}, i, x_i^{j,\beta})\ \forall i \in [n], j \in [Q_i]$	$\mathsf{ct}_i^j \leftarrow \mathsf{Enc}(\mathsf{msk}, i, x_i^{j,\beta})\ \forall i \in [n], j \in [Q_i]$
$\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}\left(\mathsf{mpk}, (\mathsf{ct}_i^j)_{i\in[n], j\in[Q_i]}\right)$	$\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}\left(\mathsf{mpk}, (\mathsf{ct}_i^j)_{i\in[n], j\in[Q_i]}\right)$
Output: β'	***Output:*** *1 if* $\beta' = \beta$*, 0 otherwise.*

where $\mathcal{A}$ *only makes queries* f *to* $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ *satisfying*

$$f(x_1^{j_1,0}, \ldots, x_n^{j_1,0}) = f(x_1^{j_1,1}, \ldots, x_n^{j_1,1})$$

for all $j_1, \ldots, j_n \in [Q_1] \times \cdots \times [Q_n]$*. For* $xx = $ one*, we require additionally that the adversary* $\mathcal{A}$ *only sends one challenge per slot, i.e. for all* $i \in [n]$*,* $Q_i = 1$*.*

The private key multi-input functional encryption $\mathcal{MIFE}$ *is xx-SEL-IND-secure if for every PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that for all* $\lambda \in \mathbb{N}$*:* $\mathsf{Adv}_{\mathcal{A}}^{\mathcal{MIFE}, xx\text{-}SEL\text{-}IND}(\lambda) = \mathsf{negl}(\lambda)$*.*

Remark 1 (winning condition). Note that the winning condition is in general not efficiently checkable because of the combinatorial explosion in the restriction on the queries.

Next, we present the simulation-based security definition for MIFE, in the setting with a single challenge ciphertext per slot.

Definition 6 (one-SEL-SIM-secure FE). *A single-input functional encryption $\mathcal{FE}$ for function $\mathcal{F}$ is one-SEL-SIM-secure if there exists a PPT simulator[8] $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Encrypt}}, \widetilde{\mathsf{KeyGen}})$ such that for every PPT adversary $\mathcal{A}$ and every $\lambda \in \mathbb{N}$, the following two distributions are computationally indistinguishable:*

Experiment $\mathbf{REAL}^{\mathcal{FE}}(\mathbf{1}^\lambda, \mathcal{A})$:	*Experiment* $\mathbf{IDEAL}^{\mathcal{FE}}(\mathbf{1}^\lambda, \mathcal{A})$:
$x \leftarrow \mathcal{A}(1^\lambda, \mathcal{F})$	$x \leftarrow \mathcal{A}(1^\lambda, \mathcal{F})$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F})$	$(\widetilde{\mathsf{mpk}}, \widetilde{\mathsf{msk}}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F})$
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{msk}, x)$	$\mathsf{ct} \leftarrow \widetilde{\mathsf{Encrypt}}(\widetilde{\mathsf{msk}})$
$\alpha \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot)}(\mathsf{mpk}, \mathsf{ct})$	$\alpha \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(\widetilde{\mathsf{mpk}}, \mathsf{ct})$
Output: α	***Output:*** α

The oracle $\mathcal{O}(\cdot)$ in the above ideal experiment has access to an oracle that provides the value $\langle \mathbf{x}, \mathbf{y} \rangle$, for each $\mathbf{y} \in \mathbb{Z}_p^m$ queried to $\mathcal{O}(\cdot)$. Then, $\mathcal{O}(\cdot)$ returns $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{y}, \langle \mathbf{x}, \mathbf{y} \rangle)$.

Namely, for every stateful adversary $\mathcal{A}$, we define

$$\mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(\lambda, \mathcal{A}) = \left| \Pr\left[\mathbf{REAL}^{\mathcal{FE}}(\mathbf{1}^\lambda, \mathcal{A}) = 1\right] - \Pr\left[\widetilde{\mathbf{IDEAL}}^{\mathcal{FE}}(\mathbf{1}^\lambda, \mathcal{A}) = 1\right] \right|,$$

and we require that for every PPT $\mathcal{A}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $\mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(\lambda, \mathcal{A}) = \mathsf{negl}(\lambda)$.

Zero vs multiple queries in private-key setting. It is convenient in our proof of security to assume that $Q_1, \ldots, Q_n \geq 1$, that is, there is at least one ciphertext for each encryption slot, which is where the technical bulk of the work lies as we would need to reason about leakage from the ideal functionality. In the setting where some $Q_i = 0$, the ideal functionality leaks nothing, and here, we can easily achieve semantic security for all of the messages being encrypted in the private key MIFE setting, via the following simple generic transformation.

Lemma 3. *Let $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ be a private key MIFE construction for n-input functions in the class $\mathcal{F}_n$, which satisfies any xx-yy-zzz MIFE security definition when the adversary receives at least one ciphertext for each encryption slot. Let $(\mathsf{Gen}_{\mathsf{SE}}, \mathsf{Enc}_{\mathsf{SE}}, \mathsf{Dec}_{\mathsf{SE}})$ be symmetric key encryption. The private key MIFE scheme $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ described in Fig. 2 satisfies xx-yy-zzz security without any restrictions on the ciphertext challenge sets.*

Proof (sketch). We consider two cases:
- Case 1: there exists some $i \in [n]$ for which $Q_i = 0$. Here, k_i and thus K is perfectly hidden from the adversary. Then, security follows readily from semantic security of $(\mathsf{Gen}_{\mathsf{SE}}, \mathsf{Enc}_{\mathsf{SE}}, \mathsf{Dec}_{\mathsf{SE}})$.
- Case 2: for all i, $Q_i \geq 1$. Here, security follows immediately from that of $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$.

$\square$

---

[8] That is, $\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Encrypt}}, \widetilde{\mathsf{KeyGen}}$ correspond respectively to the simulated $\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}$.

```
┌─────────────────────────────────────────────────────────────────────────┐
│ Setup′(1^λ, 𝓕_n):                                                          │
│     msk ← Setup(1^λ, 𝓕_n)                                                  │
│     K ← Gen(1^λ)                                                           │
│     k_1, …, k_{n-1} ←_R {0,1}^λ, k_n = (⊕_{i∈[n-1]} k_i) ⊕ K              │
│     return msk′ ← (msk, K, {k_i}_{i∈[n]})                                 │
│                                                                           │
│ Enc′(msk, i, **x**_i):                                                     │
│     parse msk′ = (msk, K, {k_i}_{i∈[n]})                                  │
│     ct ← Enc(msk, i, **x**_i)                                              │
│     ct′ ← Enc_SE(K, ct)                                                    │
│     return (k_i, ct′)                                                      │
│                                                                           │
│ KeyGen′(msk, f):                                                           │
│     return KeyGen(msk, f)                                                  │
│                                                                           │
│ Dec′(sk_f, f, ct′_1, …, ct′_n):                                           │
│     parse { ct′_i = (k_i, ct_i) }_{i∈[n]}                                 │
│     K ← ⊕_{i∈[n]} k_i                                                      │
│     { ct_i ← Dec_SE(K, ct′_i) }_{i∈[n]}                                    │
│     return Dec(sk_f, f, ct_1, …, ct_n).                                    │
└─────────────────────────────────────────────────────────────────────────┘
```
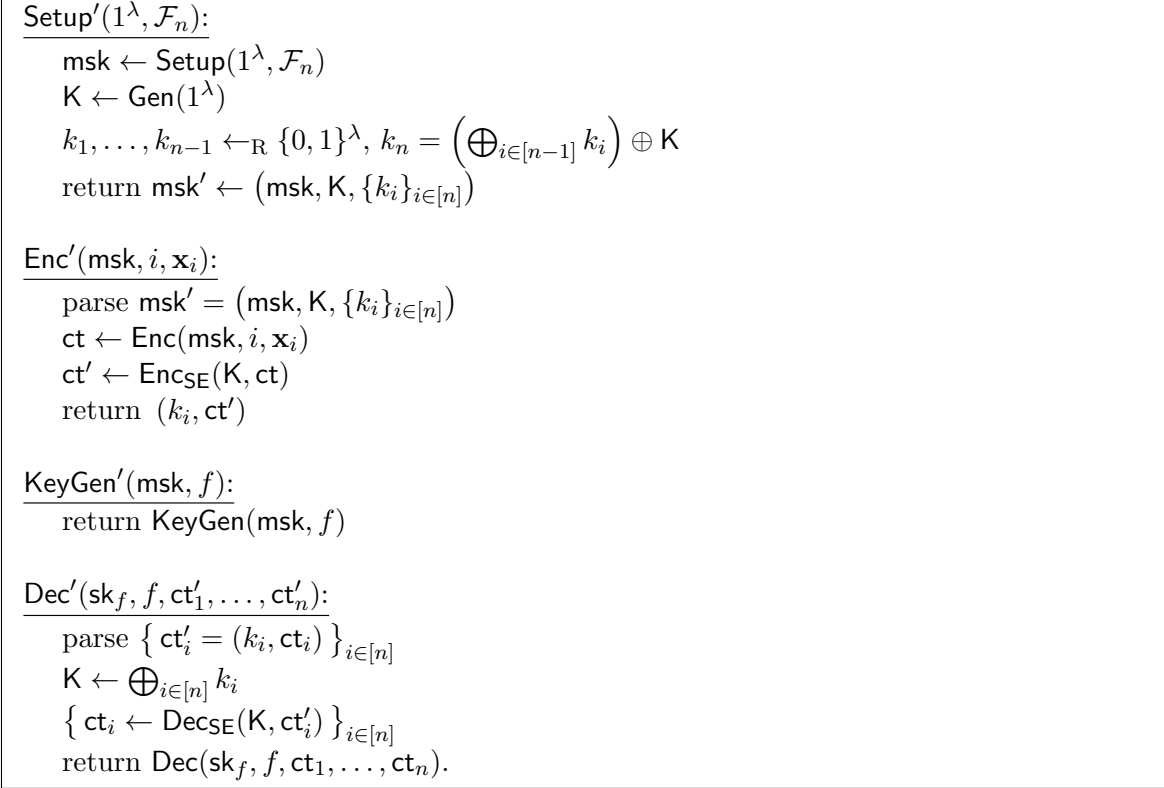
Fig. 2: Compiler from private-key MIFE with xx-yy-zzz security when $|Q_i| > 0$ for all i to private-key MIFE with xx-yy-zzz security

3.2 Inner product functionality

Multi-input Inner product. We construct a multi-input functional encryption that supports the class of multi-input bounded-norm inner product functions, which is defined as $\mathcal{F}_n^{m,B} = \{f_{\mathbf{y}_1,\dots,\mathbf{y}_n} : (\mathbb{Z}^m)^n \to \mathbb{Z}\}$ where

$$f_{\mathbf{y}_1,\dots,\mathbf{y}_n}(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{y}_i \rangle.$$

We require that the norm of the inner product of any two vector components from function and input $\langle \mathbf{x}, \mathbf{y} \rangle$ is bounded by B. This bound will determine the parameters of the bilinear map groups that we will be using in our constructions; in particular, we will choose a target group that has order $q \gg n \cdot B$. To simplify naming conventions, we will omit "bounded-norm" for the rest of the paper, but we will always refer to a multi-input inner-product functionality with this property.

Remark on leakage. Let $(\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1})_{i\in[n], j\in[Q_i]}$ be the ciphertext queries, and $\mathbf{y}_1 \| \cdots \| \mathbf{y}_n$ be a secret key query. For all slots $i \in [n]$, all $j \in [Q_i]$, and all bits $b \in \{0,1\}$, the adversary can learn

$$\langle \mathbf{x}_i^{j,b} - \mathbf{x}_i^{1,b}, \mathbf{y}_i \rangle$$

via the ideal functionality. In the IND security game, this means the adversary is restricted to queries satisfying

$$\langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle.$$

In the hybrid, we want to avoid additional constraints such as

$$\langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle.$$

4 Selectively-Secure, Private-Key MIFE for Inner Product

In this section, we present a private-key MIFE for inner product that achieves many-SEL-IND security. We use a pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ of prime order q, where q is a prime of $\Theta(\lambda)$ bits. Our construction relies on the k-Lin Assumption in $\mathbb{G}_1$ and in $\mathbb{G}_2$ and is shown in Fig. 10.

We present our construction in two steps:

- first, in Section 4.1, we show how to construct a selectively-secure MIFE scheme starting from a single-input one-SEL-SIM scheme that satisfies some additional structural properties.
- Then, in Section 4.2, we show how to instantiate the underlying single-input scheme (cf. Fig. 15, whose one-SEL-SIM security is proven in Appendix A, Theorem 5), and we present a self-contained description of the scheme in Fig. 10.

We refer the reader to Section 1.1 for an overview of the construction.

4.1 Selectively-secure, multi-input scheme from single-input scheme

Main construction. We build a private key multi-input FE $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ for the class $\mathcal{F}_n^{m,B}$, starting from a private key one-SEL-SIM secure, single-input FE $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ for the class $\mathcal{F}_1^{m+k,B}$. We present our construction in Fig. 3.

Correctness. Correctness follows readily from the correctness of the underlying scheme and the equation:

$$\langle \mathbf{x}_1 \| \cdots \| \mathbf{x}_n, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n \rangle = \left(\sum_{i=1}^{n} \langle \mathbf{x}_i \| \mathbf{z}_i, \mathbf{y}_i \| \mathbf{r} \rangle \right) - \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r} \rangle.$$

Finally, we use the fact that $\langle \mathbf{x}_1 \| \cdots \| \mathbf{x}_n, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n \rangle \mod q = \langle \mathbf{x}_1 \| \cdots \| \mathbf{x}_n, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n \rangle$, since for all slots $i \in [n]$, we have $\langle \mathbf{x}_i, \mathbf{y}_i \rangle \leq B$, and $q > Bn$.

Additional requirements. The construction and the analysis requires that $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ satisfies the following structural properties:

- The scheme can be instantiated over $\mathbb{G}_1$, where the ciphertext is a vector $[\mathbf{c}]_1$ over $\mathbb{G}_1$ and the secret key is a vector $\mathbf{d}_i$ over $\mathbb{Z}_q$.
- Enc is linearly homomorphic and public-key. More specifically, we only require that, given $\mathsf{mpk}, \mathsf{Enc}(\mathsf{msk}, \mathbf{x}), \mathbf{x}'$, we can generate a fresh random encryption of $\mathbf{x} + \mathbf{x}'$, i.e. $\mathsf{Enc}(\mathsf{msk}, \mathbf{x} + \mathbf{x}')$. This property is used in the proof of Lemma 8 and Lemma 9.
- For correctness, Dec should be linear in its inputs $(\mathbf{d}, \mathbf{y})$ and $\mathbf{c}$, so that $\mathsf{Dec}([\mathbf{d}]_2, [\mathbf{y}]_2, [\mathbf{c}]_1) = [\mathsf{Dec}(\mathbf{d}, \mathbf{y}, \mathbf{c})]_T \in \mathbb{G}_T$ can be computed using a pairing.
- For an efficient MIFE decryption, Dec must work without any restriction on the norm of the output as long as the output is in the exponent.

$$
\boxed{
\begin{aligned}
&\underline{\mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B}):}\\
&\quad (\mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B}), i = 1, \ldots, n\\
&\quad \mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k, i = 1, \ldots, n\\
&\quad (\mathsf{mpk}, \mathsf{msk}) := \left(\left\{ \mathsf{mpk}_i \right\}_{i \in [n]}, \left\{ \mathsf{msk}_i, \mathbf{z}_i \right\}_{i \in [n]} \right)\\
&\quad \text{return } (\mathsf{mpk}, \mathsf{msk})\\[4pt]
&\underline{\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i):}\\
&\quad \text{return } \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i \| \mathbf{z}_i)\\[4pt]
&\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):}\\
&\quad \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k\\
&\quad \mathbf{d}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}_i, \mathbf{y}_i \| \mathbf{r}), i = 1, \ldots, n\\
&\quad z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r} \rangle\\
&\quad \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \left(\left\{ [\mathbf{d}_i]_2 \right\}_{i \in [n]}, [\mathbf{r}]_2, [z]_T \right)\\
&\quad \text{return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}\\[4pt]
&\underline{\mathsf{Dec}'((\left\{ [\mathbf{d}_i]_2 \right\}_{i \in [n]}, [\mathbf{r}]_2, [z]_T), \mathbf{y}_1 \| \cdots \| \mathbf{y}_n, \mathsf{ct}_1, \ldots, \mathsf{ct}_n):}\\
&\quad [a_i]_T \leftarrow \mathsf{Dec}([\mathbf{d}_i]_2, [\mathbf{y}_i \| \mathbf{r}_i]_2, \mathsf{ct}_i), i = 1, \ldots, n\\
&\quad \text{return the discrete log of } \left(\prod_{i=1}^n [a_i]_T \right) / [z]_T
\end{aligned}
}
$$

Fig. 3: Multi-input functional encryption scheme $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ for the class $\mathcal{F}_n^{m,B}$. $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ refers to the single-input functional encryption scheme for the class $\mathcal{F}_1^{m+k,B}$.

– Let $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$ be the stateful simulator for the one-SEL-SIM security of the single-input inner-product FE scheme. We require that $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \cdot, \cdot)$ is linear in its inputs $(\mathbf{y}, a)$, so that we can compute $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, [\mathbf{y}]_2, [a]_2) = [\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{y}, a)]_2$. This property is used in the proof of Lemma 5.

Remark 2 (notation). We use subscripts and superscripts for indexing over multiple copies, and never for indexing over positions or exponentiation. Concretely, the j'th ciphertext query in slot i is $\mathbf{x}_i^j$.

Security. Theorem 1 and Theorem 2 below, together with the fact that one-SEL-SIM security implies one-SEL-IND security, which itself implies many-SEL-IND security for a public-key FE, such as $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen})$ used in the construction presented in Fig. 3, implies the many-SEL-IND security of the MIFE $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}')$.

Theorem 1 (one-SEL-IND security of $\mathcal{MIFE}$). *Suppose the single-input FE $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ is one-SEL-SIM secure, and that the $\mathcal{D}_k$-MDDH assumption holds in $\mathbb{G}_2$. Then, the multi-input FE $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ is one-SEL-IND-secure.*

game	ct_i:	$\{\mathbf{d}_i\}_{i\in[n]}$ in $\mathsf{sk}_\mathbf{y}$:	z in $\mathsf{sk}_\mathbf{y}$:	justification/remark
0	$\mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^\beta \| \mathbf{z}_i)$	$\mathsf{KeyGen}(\mathsf{msk}_i, \mathbf{y}_i \| \mathbf{r})$	$z = \langle \mathbf{z}_1 + \ldots + \mathbf{z}_n, \mathbf{r} \rangle$	one-SEL-IND security game
1	$\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i)$	$\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta \| \mathbf{z}_i, \mathbf{y}_i \| \mathbf{r} \rangle)$	$z = \langle \mathbf{z}_1 + \ldots + \mathbf{z}_n, \mathbf{r} \rangle$	one-SEL-SIM security of $\mathcal{FE}$
2	$\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i)$	$\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle + \tilde{z}_i)$	$z = \tilde{z}_1 + \ldots + \tilde{z}_n$	$\mathcal{D}_k$-MDDH
3	$\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i)$	$\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \tilde{z}_i)$	$z = \tilde{z}_1 + \ldots + \tilde{z}_n - \sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle$	inf. theoretic

Fig. 4: Sequence of games for the proof of Theorem 1. Here, for any slot $i \in [n]$, ct_i refers to the challenge ciphertext computed by oracle $\mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$, $\mathbf{d}_i$ and z refers to the vectors computed by the oracle $\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n)$ as part of $\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}$, and $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$ is the stateful simulator for the one-SEL-SIM security of $\mathcal{FE}$.

That is, we show that our multi-input FE is selectively secure when there is only a single challenge ciphertext.

Proof (of Theorem 1). We proceed via a series of Game_i for $i \in \{0, \ldots, 3\}$, described in Fig. 5. The transitions are summarized in Fig. 4. Let $\mathcal{A}$ be a PPT adversary, and $\lambda \in \mathbb{N}$ be the security parameter.

Game$_0$: is the experiment **one-SEL-IND**$^{\mathcal{MIFE}}$ (see Definition 5).

Game$_1$: we replace $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc})$ by the efficient simulator $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}})$, using the one-SEL-SIM security of $\mathcal{FE}$, via a hybrid argument across all slots $i \in [n]$ (cf Lemma 4).

Lemma 4 (Game$_0$ to Game$_1$). *There exists a PPT adversary $\mathcal{B}_1$ such that*

$$\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \leq n \cdot \mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(1^\lambda, \mathcal{B}_1).$$

Proof. In Game$_1$, we replace $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen})$ by $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{KeyGen}})$, which is a PPT simulator whose existence is ensured by the one-SEL-SIM security of $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc})$ (see Definition 6). A complete description of Games$_0$ and Game$_1$ is given in Fig. 5.

We use a hybrid argument, which involves hybrid Game$_{0.\ell}$ for $\ell \in \{0, \ldots, n\}$, defined in Fig. 6, and we use $\mathsf{Adv}_{0.\ell}(\lambda, \mathcal{A})$ to denote $\Pr[\mathrm{Game}_{0.\ell}(\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and Game$_{0.\ell}$. Notice that Game$_0$ and Game$_1$ are identical to Game$_{0.0}$ and Game$_{0.n}$, respectively. For any $\ell \in [n]$, we build a PPT adversary $\mathcal{B}_{0.\ell}$ such that

$$\mathsf{Adv}_{0.\ell-1}(\mathcal{A}) - \mathsf{Adv}_{0.\ell}(\mathcal{A}) \leq \mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(1^\lambda, \mathcal{B}_{0.\ell}).$$

Adversary $\mathcal{B}_{0.\ell}$ is described in Fig. 7.

$$
\boxed{
\begin{array}{l|l}
\begin{array}{l}
\underline{\mathrm{Game}_0(1^\lambda, \mathcal{A}):} \\[4pt]
\beta \leftarrow_{\mathrm{R}} \{0,1\},\ \mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k \\
\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n) \\
(\mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_n) \\
\mathsf{mpk} := \{\mathsf{mpk}_i\}_{i\in[n]},\ \mathsf{msk} := \{\mathsf{msk}_i, \mathbf{z}_i\}_{i\in[n]} \\
\mathsf{ct}_i := \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^\beta \| \mathbf{z}_i) \\
\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk},\cdot)}\big(\mathsf{mpk}, (\mathsf{ct}_i)_{i\in[n]}\big) \\
\textbf{Output: } 1 \text{ if } \beta' = \beta,\ 0 \text{ otherwise.}
\end{array}
&
\begin{array}{l}
\underline{\mathrm{Game}_1(1^\lambda, \mathcal{A}):} \\[4pt]
\beta \leftarrow_{\mathrm{R}} \{0,1\},\ \mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k \\
\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n) \\
\big(\widetilde{\mathsf{mpk}}_i, \widetilde{\mathsf{msk}}_i\big) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B}) \\
\mathsf{mpk} := \{\widetilde{\mathsf{mpk}}_i\}_{i\in[n]};\ \mathsf{msk} := \{\widetilde{\mathsf{msk}}_i, \mathbf{z}_i\}_{i\in[n]} \\
\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i) \\
\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk},\cdot)}\big(\mathsf{mpk}, (\mathsf{ct}_i)_{i\in[n]}\big) \\
\textbf{Output: } 1 \text{ if } \beta' = \beta,\ 0 \text{ otherwise.}
\end{array}
\\[2pt]
\hline
\begin{array}{l}
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\[4pt]
\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k \\
\mathbf{d}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}_i, \mathbf{y}_i \| \mathbf{r}) \\
z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r} \rangle \\
\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \Big(\big\{ [\mathbf{d}_i]_2 \big\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T \Big) \\
\text{Return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
&
\begin{array}{l}
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\[4pt]
\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k \\
\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}\big(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta \| \mathbf{z}_i, \mathbf{y}_i \| \mathbf{r} \rangle\big) \\
z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r} \rangle \\
\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \Big(\big\{ [\mathbf{d}_i]_2 \big\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T \Big) \\
\text{Return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
\\[2pt]
\hline
\begin{array}{l}
\underline{\mathrm{Game}_2(1^\lambda, \mathcal{A}):} \\[4pt]
\beta \leftarrow_{\mathrm{R}} \{0,1\} \\
\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n) \\
\big(\widetilde{\mathsf{mpk}}_i, \widetilde{\mathsf{msk}}_i\big) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B}) \\
\mathsf{mpk} := \{\widetilde{\mathsf{mpk}}_i\}_{i\in[n]};\ \mathsf{msk} := \{\widetilde{\mathsf{msk}}_i\}_{i\in[n]} \\
\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i) \\
\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk},\cdot)}\big(\mathsf{mpk}, (\mathsf{ct}_i)_{i\in[n]}\big) \\
\textbf{Output: } 1 \text{ if } \beta' = \beta,\ 0 \text{ otherwise.}
\end{array}
&
\begin{array}{l}
\underline{\mathrm{Game}_3(1^\lambda, \mathcal{A}):} \\[4pt]
\beta \leftarrow_{\mathrm{R}} \{0,1\} \\
\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n) \\
\big(\widetilde{\mathsf{mpk}}_i, \widetilde{\mathsf{msk}}_i\big) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B}) \\
\mathsf{mpk} := \{\widetilde{\mathsf{mpk}}_i\}_{i\in[n]};\ \mathsf{msk} := \{\widetilde{\mathsf{msk}}_i\}_{i\in[n]} \\
\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i) \\
\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk},\cdot)}\big(\mathsf{mpk}, (\mathsf{ct}_i)_{i\in[n]}\big) \\
\textbf{Output: } 1 \text{ if } \beta' = \beta,\ 0 \text{ otherwise.}
\end{array}
\\[2pt]
\hline
\begin{array}{l}
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\[4pt]
\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k;\ \tilde{z}_1, \ldots, \tilde{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q \\
\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}\big(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle + \tilde{z}_i\big) \\
z := \tilde{z}_1 + \cdots + \tilde{z}_n \\
\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \Big(\big\{ [\mathbf{d}_i]_2 \big\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T \Big) \\
\text{Return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
&
\begin{array}{l}
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\[4pt]
\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k;\ \tilde{z}_1, \ldots, \tilde{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q \\
\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}\big(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \tilde{z}_i\big) \\
z := \tilde{z}_1 + \cdots + \tilde{z}_n - \sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle \\
\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \Big(\big\{ [\mathbf{d}_i]_2 \big\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T \Big) \\
\text{Return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
\end{array}
}
$$

Fig. 5: Game_i for $i \in \{0, \ldots, 3\}$ for the proof of Theorem 1.

-Simulation of mpk: First, $\mathcal{B}_{0,\ell}$ receives the challenge $\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}}$ from $\mathcal{A}$. Then, it picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$, $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$ for all $i \in [n]$, and sends $\mathbf{x}_\ell^\beta \| \mathbf{z}_\ell$ to the experiment it is interacting with, which
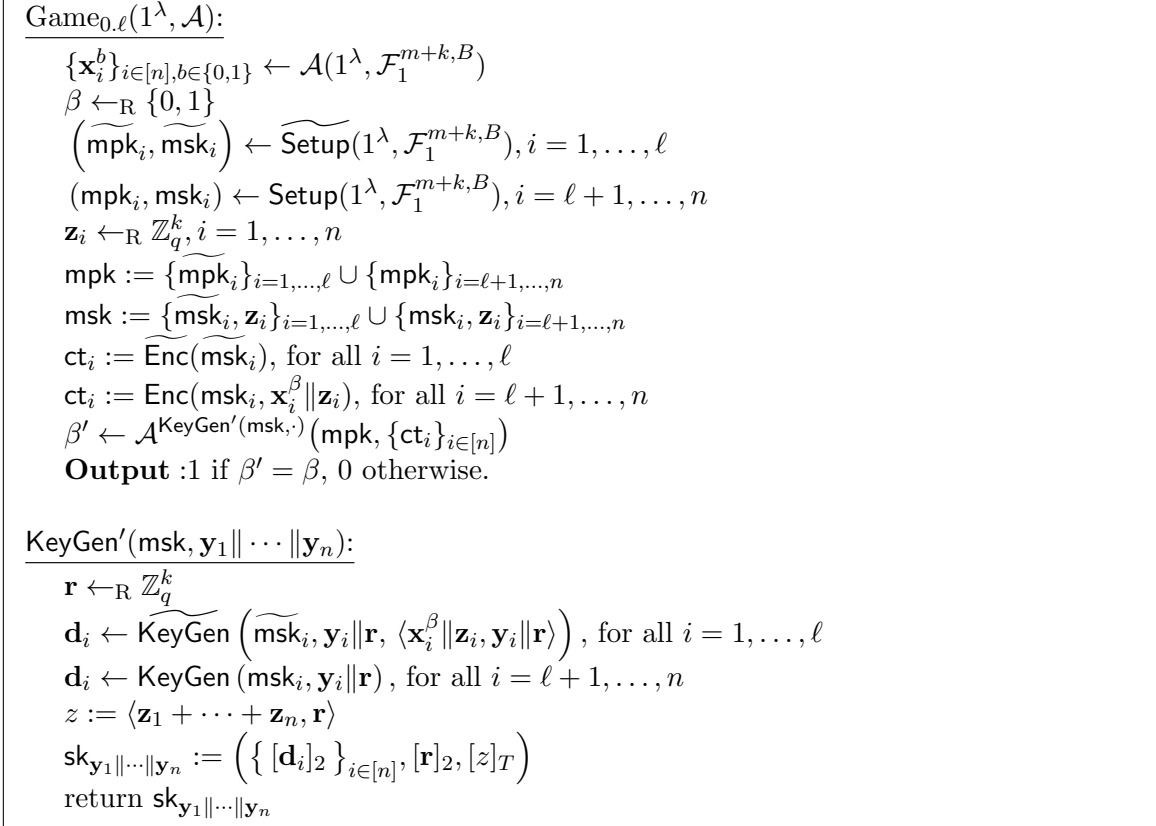
<div style="border:1px solid black; padding:10px">

$\underline{\mathrm{Game}_{0.\ell}(1^\lambda, \mathcal{A}):}$

$\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_1^{m+k,B})$

$\beta \leftarrow_{\mathrm{R}} \{0,1\}$

$\left(\widetilde{\mathsf{mpk}_i}, \widetilde{\mathsf{msk}_i}\right) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B}), i = 1, \dots, \ell$

$(\mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B}), i = \ell+1, \dots, n$

$\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k, i = 1, \dots, n$

$\mathsf{mpk} := \{\widetilde{\mathsf{mpk}_i}\}_{i=1,\dots,\ell} \cup \{\mathsf{mpk}_i\}_{i=\ell+1,\dots,n}$

$\mathsf{msk} := \{\widetilde{\mathsf{msk}_i}, \mathbf{z}_i\}_{i=1,\dots,\ell} \cup \{\mathsf{msk}_i, \mathbf{z}_i\}_{i=\ell+1,\dots,n}$

$\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}_i})$, for all $i = 1, \dots, \ell$

$\mathsf{ct}_i := \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^\beta \| \mathbf{z}_i)$, for all $i = \ell+1, \dots, n$

$\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk},\cdot)}\left(\mathsf{mpk}, \{\mathsf{ct}_i\}_{i\in[n]}\right)$

Output :1 if $\beta' = \beta$, 0 otherwise.

$\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):}$

$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$

$\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}\left(\widetilde{\mathsf{msk}_i}, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta \| \mathbf{z}_i, \mathbf{y}_i \| \mathbf{r}\rangle\right)$, for all $i = 1, \dots, \ell$

$\mathbf{d}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}_i, \mathbf{y}_i \| \mathbf{r})$, for all $i = \ell+1, \dots, n$

$z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r}\rangle$

$\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \left(\left\{[\mathbf{d}_i]_2\right\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T\right)$

return $\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}$

</div>

Fig. 6: Description of $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}')$ defining game $0.\ell$ for the proof of Lemma 4.

is either $\mathbf{REAL}^{\mathcal{FE}}$ or $\widetilde{\mathbf{IDEAL}}^{\mathcal{FE}}$. Then, $\mathcal{B}_{0.\ell}$ receives mpk'_ℓ, and a ciphertext ct, which are either of the form $\mathsf{mpk}'_\ell := \mathsf{mpk}_\ell$, where $(\mathsf{msk}_\ell, \mathsf{mpk}_\ell) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B})$, and $\mathsf{ct} := \mathsf{Enc}(\mathsf{msk}_\ell, \mathbf{x}_\ell^\beta \| \mathbf{z}_\ell)$ if $\mathcal{B}_{0.\ell}$ is interacting with the experiment $\mathbf{REAL}^{\mathcal{FE}}$; or of the form $\mathsf{mpk}'_\ell := \widetilde{\mathsf{mpk}_\ell}$, where $(\widetilde{\mathsf{msk}_\ell}, \widetilde{\mathsf{mpk}_\ell}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B})$, $\mathsf{ct} := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}_\ell})$ if $\mathcal{B}_{0.\ell}$ is interacting with the experiment $\widetilde{\mathbf{IDEAL}}^{\mathcal{FE}}$. It samples $(\widetilde{\mathsf{mpk}_i}, \widetilde{\mathsf{msk}_i}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B})$ for $i = 1, \dots, \ell-1$, $(\mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B})$ for $i = \ell+1, \dots, n$, and returns $\mathsf{mpk} := (\widetilde{\mathsf{mpk}_1}, \dots, \widetilde{\mathsf{mpk}_{\ell-1}}, \mathsf{mpk}'_\ell, \mathsf{mpk}_{\ell+1}, \dots, \mathsf{mpk}_n)$ to $\mathcal{A}$.

-Simulation of ct_i: $\mathcal{B}_{0.\ell}$ computes $\mathsf{ct}_i := \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^\beta \| \mathbf{z}_i)$ for all $i < \ell$ (note that $\mathcal{B}_{0.\ell}$ can do so since it knows msk_i, $\mathbf{x}_i^\beta$, and $\mathbf{z}_i$), and computes $\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}_i})$ for all $i > \ell$ (again, $\mathcal{B}_{0.\ell}$ can do so since it knows $\widetilde{\mathsf{msk}_i}$). Finally, $\mathcal{B}_{0.\ell}$ sets $\mathsf{ct}_\ell := \mathsf{ct}$ and returns $\{\mathsf{ct}_i\}_{i\in[n]}$ to $\mathcal{A}$.

-Simulation of $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$: For each query $\mathbf{y}_1 \| \dots \| \mathbf{y}_n$ that $\mathcal{A}$ makes to $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$, $\mathcal{B}_{0.\ell}$ picks $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and computes $\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}_i}, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta \| \mathbf{z}_i, \mathbf{y}_i \| \mathbf{r}\rangle)$ for $i = 1, \dots, \ell-1$, $\mathbf{d}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}_i, \mathbf{y}_i \| \mathbf{r})$ for $i = \ell+1, \dots, n$. Then it computes $\mathbf{d}_\ell$ by querying the oracle it has access to, which is $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ in the experiment $\mathbf{REAL}^{\mathcal{FE}}$, or $\mathcal{O}(\cdot)$ in the experiment $\widetilde{\mathbf{IDEAL}}^{\mathcal{FE}}$, on input $\mathbf{y}_\ell \| \mathbf{r}$. Then, it computes $z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r}\rangle$ and it returns $\mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n} := \left(\{[\mathbf{d}_i]_2\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T\right)$.

16

$$\underline{\mathcal{B}_{0.\ell}\left(1^\lambda, \{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}}\right):}$$

-Simulation of mpk:

$\beta \leftarrow_{\mathrm{R}} \{0,1\}$

$\left(\widetilde{\mathsf{mpk}}_i, \widetilde{\mathsf{msk}}_i\right) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B}), \text{ for } i < \ell$

$(\mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B}), i > \ell$

$\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k \text{ for } i = 1, \ldots, n$

sends $\mathbf{x}_\ell^\beta \| \mathbf{z}_\ell$ to Exp, receives mpk_ℓ' and ct

$\mathsf{mpk} := (\widetilde{\mathsf{mpk}}_1, \ldots, \widetilde{\mathsf{mpk}}_{\ell-1}, \mathsf{mpk}_\ell', \mathsf{mpk}_{\ell+1}, \ldots, \mathsf{mpk}_n)$

return mpk

-Simulation of ct_i:

For $i < \ell$: $\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i)$

For $i > \ell$: $\mathsf{ct}_i := \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^\beta \| \mathbf{z}_i)$

For $i = \ell$: $\mathsf{ct}_i := \mathsf{ct}$

-Simulation of $\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n)$:

$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$

For $i < \ell$: $\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}\left(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \langle \mathbf{x}_i^\beta \| \mathbf{z}_i, \mathbf{y}_i \| \mathbf{r} \rangle \right)$

For $i > \ell$: $\mathbf{d}_i \leftarrow \mathsf{KeyGen}\left(\mathsf{msk}_i, \mathbf{y}_i \| \mathbf{r}\right)$

For $i = \ell$: $\mathbf{d}_i \leftarrow \mathsf{KeyGenO}(\mathbf{y}_i \| \mathbf{r})$

$z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r} \rangle$

return $\left(\left\{ [\mathbf{d}_i]_2 \right\}_{i\in[n]}, \mathbf{r}, [z]_T\right)$

Fig. 7: Description of $\mathcal{B}_{0.\ell}$, adversary which distinguishes between case 1: $\mathsf{Exp} = \mathbf{REAL}^{\mathcal{FE}}$, $\mathsf{KeyGenO}(\cdot) = \mathsf{KeyGen}(\mathsf{msk}, \cdot)$, $\mathsf{mpk}_\ell' := \mathsf{mpk}_\ell$, where $(\mathsf{msk}_\ell, \mathsf{mpk}_\ell) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B})$, and $\mathsf{ct} := \mathsf{Enc}(\mathsf{msk}_\ell, \mathbf{x}_\ell^\beta \| \mathbf{z}_\ell)$; and case 2: $\mathsf{Exp} = \widetilde{\mathbf{IDEAL}}^{\mathcal{FE}}$, $\mathsf{KeyGenO}(\cdot) = \mathcal{O}(\cdot)$, $\mathsf{mpk}_\ell' := \widetilde{\mathsf{mpk}}_\ell$, where $(\widetilde{\mathsf{msk}}_\ell, \widetilde{\mathsf{mpk}}_\ell) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B})$, $\mathsf{ct} := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_\ell)$.

Finally, $\mathcal{B}_{0.\ell}$ outputs 1 if $\mathcal{A}$ outputs 1, 0 otherwise. It is clear that when $\mathcal{B}_{0.\ell}$ interacts with the experiment $\mathbf{REAL}^{\mathcal{FE}}$, it simulates the Game 0, whereas it simulates the Game 1 when it interacts with $\mathbf{IDEAL}^{\mathcal{FE}}$. Therefore,

$$\mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(\lambda, 1^\lambda, \mathcal{B}_{0.\ell})$$
$$= \left| \Pr\left[\mathbf{REAL}^{\mathcal{FE}}(1^\lambda, \mathcal{B}_{0.\ell}) = 1\right] - \Pr\left[\mathbf{IDEAL}^{\mathcal{FE}}(1^\lambda, \mathcal{B}_{0.\ell}) = 1\right] \right|$$
$$= |\mathsf{Adv}_{0.\ell-1}(\mathcal{A}) - \mathsf{Adv}_{0.\ell}(\mathcal{A})|$$

Summing up for all $\ell \in [n]$, we obtain the lemma. $\qquad\square$

Game$_2$: we replace the values $\langle \mathbf{z}_i, \mathbf{r} \rangle$ used by $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$ to $\tilde{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, for all slots $i \in [n]$, using the $\mathcal{D}_k$-MDDH assumption in $\mathbb{G}_2$ (cf Lemma 5).

Lemma 5 (Game$_1$ to Game$_2$). *There exists a PPT adversary $\mathcal{B}_2$ such that:*

$$\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_2,\mathcal{B}_2}^{\mathcal{U}_k\text{-MDDH}}(\lambda) + \frac{1}{q-1}.$$

Proof. Here, we switch $\{[\mathbf{r}]_2, [\langle \mathbf{z}_i, \mathbf{r} \rangle]_2\}_{i \in [n]}$ used by $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ to $\{[\mathbf{r}]_2, [\tilde{z}_i]_2\}_{i \in [n]}$, where for all $i \in [n]$, $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, $\tilde{z}_1, \ldots, \tilde{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$. This is justified by the fact that $[\mathbf{r}^\top \| \langle \mathbf{z}_1, \mathbf{r} \rangle \| \cdots \| \langle \mathbf{z}_n, \mathbf{r} \rangle]_2 \in \mathbb{G}_2^{1 \times (k+n)}$ is identically distributed to $[\mathbf{r}^\top \mathbf{U}^\top]_2$ where $\mathbf{U} \leftarrow_{\mathrm{R}} \mathcal{U}_{k+n,k}$ (wlog. we assume that the upper k rows of $\mathbf{U}$ are full rank), which is indistinguishable from a uniformly random vector over $\mathbb{G}_2^{1 \times (k+n)}$, that is, of the form: $[\mathbf{r} \| \tilde{z}_1 \| \cdots \| \tilde{z}_n]_2$, according to the $\mathcal{U}_{k+n,k}$-MDDH assumption. To do the switch simultaneously for all calls to KeyGen, that is, to switch $\{[\mathbf{r}^j]_2, [\langle \mathbf{z}_i, \mathbf{r}^j \rangle]_2\}_{i \in [n], j \in [Q_0]}$ to $\{[\mathbf{r}^j]_2, [\tilde{z}_i^j]_2\}_{i \in [n], j \in [Q_0]}$, where Q_0 denotes the number of calls to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, and for all $i \in [n]$, $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, $\tilde{z}_1^j, \ldots, \tilde{z}_n^j \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and for all $j \in [Q_0]$, $\mathbf{r}^j \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, we use the Q_0-fold $\mathcal{U}_{k+n,k}$-MDDH assumption. Namely, we build a PPT adversary $\mathcal{B}_2'$ such that $\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_2,\mathcal{B}_2'}^{n\text{-fold } \mathcal{U}_{Q_0,k}\text{-MDDH}}(\lambda)$. This, together with Lemma 1 ($\mathcal{U}_k$-MDDH $\Rightarrow$ n-fold $\mathcal{U}_{Q_0,k}$-MDDH), implies the lemma.

-Simulation of mpk: Upon receiving an Q_0-fold $\mathcal{U}_{k+n,k}$-MDDH challenge

$$\left(\mathcal{PG}, [\mathbf{U}]_2 \in \mathbb{G}_2^{(k+n) \times k}, \left[\mathbf{h}^1 \| \cdots \| \mathbf{h}^{Q_0} \right]_2 \in \mathbb{G}_2^{(k+n) \times Q_0} \right),$$

and the challenge $\{\mathbf{x}_i^b\}_{i \in [n], b \in \{0,1\}}$ from $\mathcal{A}$, $\mathcal{B}_1'$ picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$, samples $(\widetilde{\mathsf{mpk}}_i, \widetilde{\mathsf{msk}}_i) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}_1^{m+k,B})$ for $i \in [n]$, and returns $\mathsf{mpk} := (\widetilde{\mathsf{mpk}}_1, \ldots, \widetilde{\mathsf{mpk}}_n)$ to $\mathcal{A}$.

-Simulation of ct$_i$: $\mathcal{B}_2'$ computes $\mathsf{ct}_i := \widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}}_i)$ for all $i \in [n]$, which it can do since it knows $\widetilde{\mathsf{msk}}_i$, and returns $\{\mathsf{ct}_i\}_{i \in [n]}$ to $\mathcal{A}$.

-Simulation of KeyGen$'$(msk, $\cdot$): On the j'th query $\mathbf{y}_1 \| \cdots \| \mathbf{y}_n$ of $\mathcal{A}$ to KeyGen', $\mathcal{B}_2'$ sets $[\mathbf{r}^j]_2 := [\overline{\mathbf{h}^j}]_2$, where $\overline{\mathbf{h}^j} \in \mathbb{Z}_q^k$ denotes the k-upper components of $\mathbf{h}^j \in \mathbb{Z}_q^{k+n}$, and for each $i \in [n]$, computes $[\mathbf{d}_i]_2 := [\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}^j, \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle + \mathbf{h}_{k+i}^j)]_2$, where $\mathbf{h}_{k+i}^j$ denotes the $k + i$'th coordinate of the vector $\mathbf{h}^j \in \mathbb{Z}_p^{k+n}$. Here we rely on the fact that $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \cdot, \cdot)$ is linear in its inputs $(\mathbf{y}, a)$, so that we can compute $\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, [\mathbf{y}]_2, [a]_2) = [\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{y}, a)]_2$. Note that when $\left[\mathbf{h}^1 \| \cdots \| \mathbf{h}^{Q_0} \right]_2$ is a real MDDH challenge, $\mathcal{B}_2'$ simulate Game$_1$, whereas it simulates Game$_2$ when $\left[\mathbf{h}^1 \| \cdots \| \mathbf{h}^{Q_0} \right]_2$ is uniformly random over $\mathbb{G}_1^{(k+n) \times Q_0}$. $\square$

Game$_3$: here the values $\mathbf{d}_i$ for $i \in [n]$, and z, computed by $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$, are of the form: $\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}} \left(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i \| \mathbf{r}, \boxed{\tilde{z}_i} \right)$, and $z := \tilde{z}_1 + \cdots + \tilde{z}_n - \boxed{\sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle}$. In Lemma 6, we prove that Game$_3$ and Game$_2$ are perfectly indistinguishable, using a statistical argument that crucially relies on the fact that Game$_3$ and Game$_2$ are selective. In Lemma 7, we prove that no adversary can win Game$_3$, using the restriction on the queries to $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$ and the challenge $\{\mathbf{x}_i^b\}_{i \in [n]}$ imposed by the ideal functionality.

18

Lemma 6 (Game$_2$ to Game$_3$). $\mathsf{Adv}_2(\mathcal{A}) = \mathsf{Adv}_3(\mathcal{A})$.

Proof. Here, we use the fact that for all $\mathbf{y}_1\|\cdots\|\mathbf{y}_n \in (\mathbb{Z}_q^m)^n$, for all $\{\mathbf{x}_i^b \in \mathbb{Z}_q^m\}_{i\in[n],b\in\{0,1\}}$, all $\beta \in \{0,1\}$, the following are identically distributed: $\{\tilde{z}_i\}_{i\in[n]}$ and $\{\tilde{z}_i - \boxed{\langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle}\}_{i\in[n]}$, where $\tilde{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ for all $i \in [n]$.

For each query $\mathbf{y}_1\|\cdots\|\mathbf{y}_n$, $\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)$ picks values $\tilde{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ for $i \in [n]$ that are *independent* of $\mathbf{y}_1\|\cdots\|\mathbf{y}_n$ and the challenge $\{\mathbf{x}_i^b \in \mathbb{Z}_q^m\}_{i\in[n],b\in\{0,1\}}$ (note that here we crucially rely on the fact the Game$_2$ and Game$_3$ are *selective*), therefore, using the previous fact, we can switch $\tilde{z}_i$ to $\tilde{z}_i - \boxed{\langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle}$ without changing the distribution of the game. This way, $\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)$ computes $\mathbf{d}_i \leftarrow \widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}_i, \mathbf{y}_i\|\mathbf{r}, \tilde{z}_i)$ for all $i \in [n]$, and $z := \tilde{z}_1 + \ldots + \tilde{z}_n - \sum_{i=1}^n \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle$, as in Game$_3$.

$\square$

Lemma 7 (Game$_3$). $\mathsf{Adv}_3(\mathcal{A}) = 0$.

Proof. We use the fact that for all $i \in [n]$, the query $(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$ to Enc' (recall that there can be at most one query per slot $i \in [n]$), and for all queries $\mathbf{y}_1\|\cdots\|\mathbf{y}_n$ to KeyGen', by definition of the security game, we have:

$$\sum_{i=1}^n \langle \mathbf{x}_i^0, \mathbf{y}_i \rangle = \sum_{i=1}^n \langle \mathbf{x}_i^1, \mathbf{y}_i \rangle.$$

Therefore, for each call to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, the value z, which is of the form $z := \sum_i \tilde{z}_i - \sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle$, is independent of β. Since the challenge ciphertext and the public key are also independent of β, we have $\mathsf{Adv}_3(\mathcal{A}) = 0$. $\square$

Summing up, we proved that for all security parameter $\lambda \in \mathbb{N}$ and all PPT adversaries $\mathcal{A}$, the following holds.

- In Lemma 4, we show that there exists a PPT adversary $\mathcal{B}_1$ such that $\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \leq n \cdot \mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(1^\lambda, \mathcal{B}_1)$.
- In Lemma 5, we show that there exists a PPT adversary $\mathcal{B}_2$ such that $\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A}) \leq \mathsf{Adv}^{\mathcal{U}_k\text{-}\mathrm{MDDH}}_{\mathbb{G}_2, \mathcal{B}_2}(\lambda) + \frac{1}{q-1}$.
- In Lemma 6, we show that $\mathsf{Adv}_2(\mathcal{A}) = \mathsf{Adv}_3(\mathcal{A})$.
- In Lemma 7, we show that $\mathsf{Adv}_3(\mathcal{A}) = 0$.

Putting everything together, we obtain:

$$\mathsf{Adv}_0(\mathcal{A}) \leq n \cdot \mathsf{Adv}^{\mathcal{FE}, one\text{-}SEL\text{-}SIM}(1^\lambda, \mathcal{B}_0) + \mathsf{Adv}^{\mathcal{U}_k\text{-}\mathrm{MDDH}}_{\mathbb{G}_2, \mathcal{B}_2}(\lambda) + \frac{1}{q-1}.$$

By Definition 6, $\mathsf{Adv}_0(\mathcal{A}) = \mathsf{Adv}^{\mathcal{MIFE}, one\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{A})$. Therefore, by the one-SEL-SIM security of $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen})$ and the $\mathcal{D}_k$-MDDH assumption in $\mathbb{G}_2$, $\mathsf{Adv}^{\mathcal{MIFE}, one\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{A})$ is a negligible function of λ. $\square$

Remark 3 (decryption capabilities). As a sanity check, we note that the simulated secret keys will correctly decrypt a simulated ciphertext. However, unlike schemes proven secure via the standard dual system encryption methodology [26], a simulated secret key will incorrectly decrypt a normal ciphertext. This is not a problem because we are in the private-key setting, so a distinguisher will not be able to generate normal ciphertexts by itself.

Remark 4 (why a naive argument is inadequate). We cannot afford to do a naive hybrid argument across the n slots for the challenge ciphertext as it would introduce extraneous restrictions on the adversary's queries. Concretely, suppose we want to use a hybrid argument to switch from encryptions of $\mathbf{x}_1^0, \mathbf{x}_2^0$ in game 0 to those of $\mathbf{x}_1^1, \mathbf{x}_2^1$ in game 2 with an intermediate hybrid that uses encryptions of $\mathbf{x}_1^1, \mathbf{x}_2^0$ in Game$_1$. To move from game 0 to game 1, the adversary's query $\mathbf{y}_1\|\mathbf{y}_2$ must satisfy $\langle \mathbf{x}_1^0\|\mathbf{x}_2^0, \mathbf{y}_1\|\mathbf{y}_2\rangle = \langle \mathbf{x}_1^1\|\mathbf{x}_2^0, \mathbf{y}_1\|\mathbf{y}_2\rangle$, which implies the extraneous restriction $\langle \mathbf{x}_1^0, \mathbf{y}_1\rangle = \langle \mathbf{x}_2^1, \mathbf{y}_1\rangle$.

As described in the proof above, we overcome the limitation by using simulation-based security. Note that what essentially happens in the first slot in our proof is as follows (for $k = 1$, that is, DDH): we switch from $\mathsf{Enc}(\mathsf{msk}_1, \mathbf{x}_1^0\|z_1)$ to $\mathsf{Enc}(\mathsf{msk}_1, \mathbf{x}_1^1\|z_1)$ while giving out a secret key which contains $\mathsf{KeyGen}(\mathsf{msk}_1, \mathbf{y}_1\|r^1), [r^1]_2$. Observe that

$$\langle \mathbf{x}_1^0\|z_1, \mathbf{y}_1\|r^1\rangle = \langle \mathbf{x}_1^0, \mathbf{y}_1\rangle + z_1 r^1, \quad \langle \mathbf{x}_1^1\|z_1, \mathbf{y}_1\|r^1\rangle = \langle \mathbf{x}_1^1, \mathbf{y}_1\rangle + z_1 r^1$$

may not be equal, since we want to avoid the extraneous restriction $\langle \mathbf{x}_1^0, \mathbf{y}_1\rangle = \langle \mathbf{x}_2^1, \mathbf{y}_1\rangle$. This means that one-SEL-IND security does not provide any guarantee that the ciphertexts are indistinguishable. However, one-SEL-SIM security does provide such a guarantee, because

$$([\langle \mathbf{x}_1^0, \mathbf{y}_1\rangle + z_1 r^1]_2, [r^1]_2) \approx_c ([\langle \mathbf{x}_1^1, \mathbf{y}_1\rangle + z_1 r^1]_2, [r^1]_2)$$

via the DDH assumption in $\mathbb{G}_2$. Since the outcomes of the decryption are computationally indistinguishable, the output of the simulated ciphertext would also be computationally indistinguishable.

Theorem 2 (many-SEL-IND security of $\mathcal{MIFE}$). *Suppose the single-input FE* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *is many-SEL-IND-secure and the multi-input FE* $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ *is one-SEL-IND-secure. Then, the multi-input FE* $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ *is many-SEL-IND-secure.*

That is, we show that our multi-input FE is selectively secure in the setting with multiple challenge ciphertexts (and since our multi-input FE is a private key scheme, one-SEL-IND security does not immediately imply many-SEL-IND security).

Proof overview.

- We first switch encryptions of $\mathbf{x}_1^{1,0}, \ldots, \mathbf{x}_n^{1,0}$ to those of $\mathbf{x}_1^{1,1}, \ldots, \mathbf{x}_n^{1,1}$ in a "single shot", and for the remaining ciphertexts, we switch from an encryption of $\mathbf{x}_i^{j,0} = (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}) + \mathbf{x}_i^{1,0}$ to that of $(\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}) + \mathbf{x}_i^{1,1}$. This basically follows from the setting where there is only a single ciphertext in each slot.
- Then, we apply a hybrid argument across the slots to switch from encryptions of

$$(\mathbf{x}_i^{2,0} - \mathbf{x}_i^{1,0}) + \mathbf{x}_i^{1,1}, \ldots, (\mathbf{x}_i^{Q_i,0} - \mathbf{x}_i^{1,0}) + \mathbf{x}_i^{1,1}$$

to those of

$$(\mathbf{x}_i^{2,1} - \mathbf{x}_i^{1,1}) + \mathbf{x}_i^{1,1}, \ldots, (\mathbf{x}_i^{Q_i,1} - \mathbf{x}_i^{1,1}) + \mathbf{x}_i^{1,1}.$$

game	ct_i^j:	justification/remark
0	$\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \mathbf{x}_i^{1,0})$	many-SEL-IND security game
1	$\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \boxed{\mathbf{x}_i^{1,1}})$	one-SEL-IND security of $\mathcal{MIFE}$
1.ℓ	$\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1})$ for $i \leq \ell$ $\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}} + \mathbf{x}_i^{1,1})$ for $i > \ell$	many-SEL-IND security of $\mathcal{FE}$
2	$\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1}})$	information theoretic

Fig. 8: Sequence of games for the proof of Theorem 2. Here, for any slot $i \in [n]$, and $j \in [Q_i]$, ct_i^j refers to the challenge ciphertext computed at the oracle call: $\mathsf{Enc}'(i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1})$.

As described earlier, to carry out the latter hybrid argument, the queries must satisfy the constraint

$$\langle (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}) + \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle = \langle (\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}) + \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle \iff \langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle$$

where the latter is already imposed by the ideal functionality.

Proof (of Theorem 2). We proceed via a series of Game_i, for $i \in \{0, 1, 2\}$, and $\mathrm{Game}_{1.\ell}$ for $\ell \in \{0, \ldots, n\}$, described in Fig. 9. The transitions are summarized in Fig. 8. Let $\mathcal{A}$ be a PPT adversary, and $\lambda \in \mathbb{N}$ be the security parameter. For all $i \in \{0, 1, 2, 1.0, \ldots, 1.n\}$, we define $\mathsf{Adv}_i(\mathcal{A}) := \Pr[\mathrm{Game}_i(1^\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and Game_i.

Game$_0$: is the experiment **many-SEL-IND**$_0^{\mathcal{MIFE}}$ (see Definition 5).

Game$_1$: we change the challenge ciphertexts to $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \boxed{\mathbf{x}_i^{1,1}})$, for all slots $i \in [n]$, and all $j \in [Q_i]$, using the one-SEL-IND security of $\mathcal{MIFFE}$ (cf Lemma 8).

Lemma 8 (Game$_0$ to Game$_1$). *There exists a PPT adversary $\mathcal{B}_1$ such that*

$$|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{MIFE}, one\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{B}_1).$$

Proof. In Game 1, which is described in Fig. 9, we replace $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0}) = \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{1,0}} + (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}))$ with $\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{1,1}} + (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}))$ for all $i \in [n], j \in [Q_i]$. This is justified using the following properties:

- one-SEL-IND security of $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$;
- given $\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}$, we can maul an encryption of $\mathbf{x}_i^{1,\beta}$ under Enc' (corresponding to challenge ciphertext in slot i in the one-SEL-IND security game) into that of $\mathbf{x}_i^{1,\beta} + (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})$ (corresponding to challenge ciphertexts in slots i in Game_β).

We build $\mathcal{B}_1$, a PPT adversary such that for all $\beta \in \{0, 1\}$, when $\mathcal{B}_1$ interacts with the experiment **one-SEL-IND**$_\beta^{\mathcal{MIFE}}$ (see Definition 5), it simulates the Game_β to $\mathcal{A}$.

$\underline{\text{Game}_0(1^\lambda, \mathcal{A})\text{:}}$
$\{\mathbf{x}_i^{j,b}\}_{i\in[n],j\in[Q_i],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n^{m,B})$
$(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B})$
$\mathsf{ct}_i^j := \mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \mathbf{x}_i^{1,0})$
$\beta' \leftarrow \mathcal{A}(\mathsf{mpk}, \{\mathsf{ct}_i^j\}_{i\in[n],j\in[Q_i]})$
Return β'.

$\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)\text{:}}$
    return $\mathsf{sk}_{\mathbf{y}_1\|\cdots\|\mathbf{y}_n}$

$\underline{\text{Game}_1(1^\lambda, \mathcal{A})\text{:}}$
$\{\mathbf{x}_i^{j,b}\}_{i\in[n],j\in[Q_i],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n^{m,B})$
$(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B})$
$\mathsf{ct}_i^j := \mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \boxed{\mathbf{x}_i^{1,1}})$
$\beta' \leftarrow \mathcal{A}(\mathsf{mpk}, \{\mathsf{ct}_i^j\}_{i\in[n],j\in[Q_i]})$
Return β'.

$\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)\text{:}}$
    return $\mathsf{sk}_{\mathbf{y}_1\|\cdots\|\mathbf{y}_n}$

$\underline{\text{Game}_{1.\ell}(1^\lambda, \mathcal{A})\text{:}}$
$\{\mathbf{x}_i^{j,b}\}_{i\in[n],j\in[Q_i],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n^{m,B})$
$(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B})$
$\mathsf{ct}_i^j := \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1})$ for $i \le \ell$
$\mathsf{ct}_i^j := \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}} + \mathbf{x}_i^{1,1})$ for $i > \ell$
$\beta' \leftarrow \mathcal{A}(\mathsf{mpk}, \{\mathsf{ct}_i^j\}_{i\in[n],j\in[Q_i]})$
Return β'.

$\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)\text{:}}$
    return $\mathsf{sk}_{\mathbf{y}_1\|\cdots\|\mathbf{y}_n}$

$\underline{\text{Game}_2(1^\lambda, \mathcal{A})\text{:}}$
$\{\mathbf{x}_i^{j,b}\}_{i\in[n],j\in[Q_i],b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_n^{m,B})$
$(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B})$
$\mathsf{ct}_i^j := \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1})$
$\beta' \leftarrow \mathcal{A}(\mathsf{mpk}, \{\mathsf{ct}_i^j\}_{i\in[n],j\in[Q_i]})$
Return β'.

$\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)\text{:}}$
    return $\mathsf{sk}_{\mathbf{y}_1\|\cdots\|\mathbf{y}_n}$

Fig. 9: Game_i for $i \in \{0, 1, 2\}$, and $\text{Game}_{1.\ell}$ where $\ell \in \{0, \ldots, n\}$ for the proof of Theorem 2. The description of KeyGen' is omitted since its simulation does not change throughout the proof.

-**Simulation of mpk:** $\mathcal{B}_1$ gets the challenge $\{\mathbf{x}_i^{j,b}\}_{i\in[n],j\in[Q_i],b\in\{0,1\}}$ from $\mathcal{A}$. It sends the pair of vectors $(\mathbf{x}_1^{1,0}\|\cdots\|\mathbf{x}_n^{1,0}, \mathbf{x}_1^{1,1}\|\cdots\|\mathbf{x}_n^{1,1})$ as its selective challenge to the experiment **one-SEL-IND**$_\beta^{\mathcal{MIFE}}$, to get back mpk, which it sends to $\mathcal{A}$, and the challenge ciphertext $C_1\|\cdots\|C_n$ corresponding to $\mathsf{Enc}'(\mathsf{msk}, 1, \mathbf{x}_1^{1,\beta})\|\cdots\|\mathsf{Enc}'(\mathsf{msk}, n, \mathbf{x}_n^{1,\beta})$.

-**Simulation of ct_i^j:** For each slot $i \in [n]$, $\mathcal{B}_1$ sets $\mathsf{ct}_i^1 := C_i$. Then, $\mathcal{B}_1$ uses the fact that the single-input inner-product scheme is public-key and linearly homomorphic to generate all the remaining ciphertexts ct_i^j for $i \in [n], j \in \{2, \ldots, Q_i\}$ by combining $C_i = \mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{1,\beta}) = \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{1,\beta}\|\mathbf{z}_i)$ with $\mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}\|\mathbf{0})$ (which can be done using mpk_i). In particular, each ct_i^j will correspond to $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{1,\beta} + \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})$, which matches the challenge ciphertexts in Game_β.

-**Simulation of $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$:** $\mathcal{B}_1$ uses its KeyGen' oracle to simulate the corresponding queries by $\mathcal{A}$.

Finally, $\mathcal{B}_1$ outputs 1 if $\mathcal{A}$ outputs 1, 0 otherwise. It is clear that for all $\beta \in \{0,1\}$, when $\mathcal{B}_1$ interacts with **one-SEL-IND**$_\beta^{\mathcal{MIFE}}$, it simulates Game β to $\mathcal{A}$. Therefore,

$$\mathsf{Adv}^{\mathcal{MIFE},one\text{-}SEL\text{-}IND}(\lambda, \mathcal{B}_1)$$

$$= \left| \Pr\left[\textbf{one-SEL-IND}_0^{\mathcal{MIFE}}(1^\lambda, \mathcal{B}_1) = 1\right] - \Pr\left[\textbf{one-SEL-IND}_1^{\mathcal{MIFE}}(1^\lambda, \mathcal{B}_1) = 1\right] \right|$$

$$= |\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})|$$

$\square$

Game$_2$: we change the challenge ciphertexts to $\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1})$, for all slots $i \in [n]$, and all $j \in [Q_i]$, using the many-SEL-IND security of $\mathcal{FE}$, via a hybrid argument across all slots $i \in [n]$ (cf Lemma 9).

Lemma 9 (Game$_1$ to Game$_2$). *There exists a PPT adversary $\mathcal{B}_2$ such that* $|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq n \cdot \mathsf{Adv}^{\mathcal{FE},many\text{-}SEL\text{-}IND}(\lambda, \mathcal{B}_2)$.

Proof. In Game$_2$, which is described in Fig. 9, we replace $\mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{1,1} + \boxed{(\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})} \|\mathbf{z}_i)$ with $\mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{1,1} + \boxed{(\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1})} \|\mathbf{z}_i) = \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{j,1}\|\mathbf{z}_i)$, for all $i \in [n], j \in [Q_i]$. This is justified using the following properties:

- many-SEL-IND security of $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$;
- given $\mathbf{x}_i^{1,1}$ and $\mathbf{z}_i$, we can maul an encryption of $(\mathbf{x}_i^{j,\beta} - \mathbf{x}_i^{1,\beta})\|\mathbf{0}$ under Enc (corresponding to challenge ciphertexts in the many-SEL-IND security game) into that of $\mathbf{x}_i^{1,1} + \boxed{(\mathbf{x}_i^{j,\beta} - \mathbf{x}_i^{1,\beta})} \|\mathbf{z}_i$ (corresponding to challenge ciphertexts in Game$_{\beta+1}$);
- the fact that for each key query $\mathbf{y}_1\|\ldots\|\mathbf{y}_n$ and all $\mathbf{r}, \mathbf{z}$, we have

$$\langle \mathbf{x}_i^{1,1} + \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}\|\mathbf{z}, \mathbf{y}_i\|\mathbf{r}\rangle = \langle \mathbf{x}_i^{1,1} + \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}\|\mathbf{z}, \mathbf{y}_i\|\mathbf{r}\rangle.$$

The latter is equivalent to $\langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i\rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i\rangle$, which follows from the restriction in Section 3.2.

In more details, we use a hybrid argument, which involves hybrid Games$_{1.\ell}$ for $\ell \in \{0, \ldots, n\}$, defined in Fig. 9. Notice that Game$_1$ and Game$_2$ are identically distributed than Game$_{1.0}$ and Game$_{1.n}$, respectively. For all $\ell \in [n]$, we build a PPT adversary $\mathcal{B}_{1.\ell}$ such that for all $\beta \in \{0,1\}$, when $\mathcal{B}_{1.\ell}$ interacts with the experiment **many-SEL-IND**$_\beta^{\mathcal{FE}}$ (see Definition 5), it simulates the Game$_{1.\ell-1+\beta}$ to $\mathcal{A}$.

-Simulation of mpk: First, $\mathcal{B}_{1.\ell}$ receives the challenge $\{\mathbf{x}_i^{j,b}\}_{i\in[n],j\in[Q_i],b\in\{0,1\}}$ from $\mathcal{A}$. It picks $\mathbf{z}_i \leftarrow_\mathrm{R} \mathbb{Z}_q^k$ for all $i \in [n]$, and sends the vectors $\{\mathbf{x}_\ell^{j,0}\|\mathbf{z}_\ell, \mathbf{x}_\ell^{j,1}\|\mathbf{z}_\ell\}_{j\in[Q_\ell]}$ as its selective challenge to the experiment **many-SEL-IND**$_\beta^{\mathcal{FE}}$, to get back mpk_ℓ, and the ciphertexts $C_j := \mathsf{Enc}(\mathsf{msk}_\ell, \mathbf{x}_\ell^{j,\beta}\|\mathbf{z}_\ell)$, for all $j \in [Q_\ell]$. Then, it samples $(\mathsf{msk}_i, \mathsf{mpk}_i) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B})$ for $i \in [n], i \neq \ell$, and it sends $\mathsf{mpk} := (\mathsf{mpk}_1, \ldots, \mathsf{mpk}_n)$ to $\mathcal{A}$.

-Simulation of ct_i^j: $\mathcal{B}_{1.\ell}$ computes $\mathsf{ct}_i^j := \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{j,1}\|\mathbf{z}_i)$ for all $i < \ell$, and $\mathsf{ct}_i^j := \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \mathbf{x}_i^{1,1}\|\mathbf{z}_i)$ for all $i > \ell$ (note that $\mathcal{B}_{1.\ell}$ can do so since it knows msk_i and $\mathbf{z}_i$ for all $i \neq \ell$). Then, for all $j \in [Q_\ell]$, $\mathcal{B}_{1.\ell}$ uses $C_j := \mathsf{Enc}(\mathsf{msk}_\ell, \mathbf{x}_\ell^{j,\beta}\|\mathbf{z}_\ell)$ together with mpk_ℓ and $(-\mathbf{x}_\ell^{1,0} + \mathbf{x}_\ell^{1,1})\|\mathbf{0}$ to generate $\mathsf{ct}_\ell^j := \mathsf{Enc}(\mathsf{msk}_\ell, \mathbf{x}_\ell^{j,\beta} - \mathbf{x}_\ell^{1,0} + \mathbf{x}_\ell^{1,1}\|\mathbf{z}_i)$. This is possible since the single-input inner-product scheme is public-key and linearly homomorphic. It returns $\{\mathsf{ct}_i^j\}_{i\in[n],j\in[Q_i]}$ to $\mathcal{A}$.

-Simulation of $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$: For each query $\mathbf{y}_1\|\dots\|\mathbf{y}_n$ that $\mathcal{A}$ makes to $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$, $\mathcal{B}_{1.\ell}$ picks $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and it computes $\mathbf{d}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}_i, \mathbf{y}_i\|\mathbf{r})$ for $i \in [n], i \neq \ell$. To compute $\mathbf{d}_\ell$, it queries the oracle $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ on input $\mathbf{y}_\ell\|\mathbf{r}$. It computes $z = \langle \mathbf{z}_1\|\dots\|\mathbf{z}_n, \mathbf{r}\rangle$ and sends $\mathsf{sk}_{\mathbf{y}_1\|\dots\|\mathbf{y}_n} := \left(\{[\mathbf{d}_i]_2\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T\right)$ to $\mathcal{A}$.

Finally, $\mathcal{B}_{1.\ell}$ outputs 1 if $\mathcal{A}$ outputs 1, 0 otherwise. It is clear that for all $\beta \in \{0, 1\}$, when $\mathcal{B}_{1.\ell}$ interacts with **many-SEL-IND**$_\beta^{\mathcal{MIFE}}$, it simulates $\mathrm{Game}_{1.\ell-1+\beta}$ to $\mathcal{A}$. Therefore,

$$\mathsf{Adv}^{\mathcal{MIFE},many\text{-}SEL\text{-}IND}(\lambda, \mathcal{B}_{1.\ell})$$
$$= \left|\Pr\left[\textbf{many-SEL-IND}_0^{\mathcal{FE}}(1^\lambda, \mathcal{B}_{1.\ell}) = 1\right] - \Pr\left[\textbf{many-SEL-IND}_1^{\mathcal{FE}}(1^\lambda, \mathcal{B}_{1.\ell}) = 1\right]\right|$$
$$= |\mathsf{Adv}_{1.\ell-1}(\mathcal{A}) - \mathsf{Adv}_{1.\ell}(\mathcal{A})|$$

Summing up for all $\ell \in [n]$, we obtain the lemma. $\qquad\qquad\square$

Summing up, we proved that for all PPT adversaries $\mathcal{A}$ and all $\lambda \in \mathbb{N}$, the following holds.

- In Lemma 8, we build a PPT adversary $\mathcal{B}_1$ such that $|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{MIFE},one\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{B}_1)$.
- In Lemma 9, we build a PPT adversary $\mathcal{B}_2$ such that $|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq n \cdot \mathsf{Adv}^{\mathcal{FE},many\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{B}_2)$, using a hybrid argument across all slots $i \in [n]$.

Putting everything together, we obtain:

$$|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{MIFE},one\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{B}_1) + n \cdot \mathsf{Adv}^{\mathcal{FE},many\text{-}SEL\text{-}IND}(1^\lambda, \mathcal{B}_2).$$

By Definition 5, we have $\mathsf{Adv}^{\mathcal{MIFE},many\text{-}SEL\text{-}IND}(\lambda, \mathcal{A}) = |\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})|$. Therefore, by the one-SEL-IND security of $(\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Enc}')$ and the many-SEL-IND security of $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc})$, $\mathsf{Adv}^{\mathcal{MIFE},many\text{-}SEL\text{-}IND}(\lambda, \mathcal{A})$ is a negligible function of λ. $\qquad\square$

4.2 Putting everything together

In Fig. 10 we spell out the details of the scheme in the previous section with a concrete instantiation of the underlying single-input inner-product scheme, which is provided for completeness in Fig. 15, and whose one-SEL-SIM security is proven in Appendix A, Theorem 5.

5 Achieving Adaptive Security

In this section, we prove that the generic construction of MIFE in Fig. 3 instantiated with the single-input FE in Fig. 15 is many-AD-IND secure. A self-contained description is given in Fig. 10.

$$\boxed{\begin{aligned}
&\underline{\mathsf{Setup}(\mathbb{G}, \mathcal{F}_n^{m,B}):}\\
&\quad \mathbf{A}_1, \ldots, \mathbf{A}_n \leftarrow_R \mathcal{D}_k\\
&\quad \mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_q^{m \times (k+1)}\\
&\quad \mathbf{V}_1, \ldots, \mathbf{V}_n \leftarrow_R \mathbb{Z}_q^{k \times (k+1)}\\
&\quad \mathbf{z}_1, \ldots, \mathbf{z}_n \leftarrow_R \mathbb{Z}_q^k\\
&\quad \mathsf{mpk} := \left\{ [\mathbf{A}_i]_1, [\mathbf{W}_i \mathbf{A}_i]_1 \right\}_{i \in [n]}\\
&\quad \mathsf{msk} := \left\{ \mathbf{W}_i, \mathbf{V}_i, \mathbf{z}_i \right\}_{i \in [n]}\\
&\quad \text{return } (\mathsf{mpk}, \mathsf{msk})\\[1em]
&\underline{\mathsf{Enc}(\mathsf{msk}, i, \mathbf{x}_i \in \mathbb{Z}_q^m):}\\
&\quad \text{pick } \mathbf{s}_i \leftarrow_R \mathbb{Z}_q^k;\\
&\quad ([\mathbf{c}_i]_1, [\mathbf{c}_i']_1, [\mathbf{c}_i'']_1) := ([\mathbf{A}_i \mathbf{s}_i]_1, [\mathbf{x}_i + \mathbf{W}_i \mathbf{A}_i \mathbf{s}_i]_1, [\mathbf{z}_i + \mathbf{V}_i \mathbf{A}_i \mathbf{s}_i]_1)\\
&\quad \text{return } ([\mathbf{c}_i]_1, [\mathbf{c}_i']_1, [\mathbf{c}_i'']_1)\\[1em]
&\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n \in (\mathbb{Z}_q^m)^n):}\\
&\quad \text{pick } \mathbf{r} \leftarrow_R \mathbb{Z}_q^k\\
&\quad \mathbf{d}_i := \mathbf{W}_i^\top \mathbf{y}_i + \mathbf{V}_i^\top \mathbf{r}\\
&\quad z := \langle \mathbf{z}_1 + \cdots + \mathbf{z}_n, \mathbf{r} \rangle\\
&\quad \text{return } \left(\left\{ [\mathbf{d}_i]_2 \right\}_{i \in [n]}, [\mathbf{r}]_2, [z]_T \right)\\[1em]
&\underline{\mathsf{Dec}\left(\left(\left\{ [\mathbf{d}_i]_2 \right\}_{i \in [n]}, [\mathbf{r}]_2, [z]_T \right), \mathbf{y}_1 \| \cdots \| \mathbf{y}_n, \left\{ [\mathbf{c}_i]_1, [\mathbf{c}_i']_1, [\mathbf{c}_i'']_1 \right\}_{i \in [n]} \right):}\\
&\quad \mathbf{out} \leftarrow \left(\sum_i e([\mathbf{c}_i']_1, [\mathbf{y}_i]_2) \cdot e([\mathbf{c}_i'']_1, [\mathbf{r}]_2) / e([\mathbf{c}_i]_1, [\mathbf{d}_i]_2) \right) / [z]_T\\
&\quad \text{return discrete log of } \mathbf{out}
\end{aligned}}$$

Fig. 10: Our private-key MIFE scheme for the class $\mathcal{F}_n^{m,B}$ (self-contained description). The scheme is many-AD-IND-secure under the $\mathcal{D}_k$-MDDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$. We use $e([\mathbf{X}]_1, [\mathbf{Y}]_2)$ to denote $[\mathbf{X}^\top \mathbf{Y}]_T$.

Security definitions. Here we recall the adaptive security notions.

Definition 7 (xx-AD-IND-secure MIFE). *For every multi-input functional encryption $\mathcal{MIFE}$ for $\mathcal{F}$, every stateful adversary $\mathcal{A}$, every security parameter $\lambda \in \mathbb{N}$, and every $xx \in \{one, many\}$, the advantage of $\mathcal{A}$ defined as*

$$\begin{aligned}
&\mathsf{Adv}^{\mathcal{MIFE}, xx\text{-}AD\text{-}IND}(\lambda, \mathcal{A})\\
&= \left| \Pr\left[\mathbf{xx\text{-}AD\text{-}IND}_0^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) = 1 \right] - \Pr\left[\mathbf{xx\text{-}AD\text{-}IND}_1^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) = 1 \right] \right|\\
&= \left| 1 - 2 \cdot \Pr\left[\mathbf{xx\text{-}AD\text{-}IND}^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) = 1 \right] \right|
\end{aligned}$$

where the experiments are defined as follows:

$$\boxed{\begin{array}{l|l}
\textit{Experiment } \textbf{xx-AD-IND}_\beta^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) : & \textit{Experiment } \textbf{xx-AD-IND}^{\mathcal{MIFE}}(1^\lambda, \mathcal{A}) : \\
 & \beta \leftarrow_{\mathrm{R}} \{0, 1\} \\
(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_n) & (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}_n) \\
\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot), \mathsf{Enc}(\cdot, \cdot, \cdot)}(\mathsf{mpk}) & \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk}, \cdot), \mathsf{Enc}(\cdot, \cdot, \cdot)}(\mathsf{mpk}) \\
\textbf{Output: } \beta' & \textbf{Output: } \textit{1 if } \beta' = \beta, \textit{0 otherwise.}
\end{array}}$$

where Enc *is an oracle that on input* (i, x_i^0, x_i^1) *outputs* $\mathsf{Enc}(\mathsf{msk}, i, x_i^\beta)$, *and where* $\mathcal{A}$ *only makes queries* f *to* $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ *satisfying*

$$f(x_1^{j_1, 0}, \ldots, x_n^{j_n, 0}) = f(x_1^{j_1, 1}, \ldots, x_n^{j_n, 1})$$

for all $j_1, \ldots, j_n \in [Q_1] \times \cdots \times [Q_n]$. *For* $xx = one$, *we require additionally that* $\mathcal{A}$ *queries* $\mathsf{Enc}(i, \cdot, \cdot)$ *at most one per slot* $i \in [n]$, *that is,* $Q_i = 1$.

A private key multi-input functional encryption $\mathcal{MIFE}$ *for* $\mathcal{F}$ *is xx-AD-IND-secure if for every PPT adversary* $\mathcal{A}$, *there exists a negligible function* negl *such that for every* $\lambda \in \mathbb{N}$: $\mathsf{Adv}^{\mathcal{MIFE}, xx\text{-}AD\text{-}IND}(\lambda, \mathcal{A}) = \mathsf{negl}(\lambda)$.

Security. The security proof proceeds in three steps: first (Theorem 3), we show that the MIFE in Fig. 10 is one-AD-IND secure, that is, it is adaptively secure when there is only a single challenge ciphertext. Then (Theorem 4), we show that the generic construction of MIFE in Fig. 3 is many-AD-IND secure, if the underlying single-input FE is many-AD-IND secure, and the MIFE is one-AD-IND secure. Finally (Appendix A), we give the proof that the single-input FE in Fig. 15 is many-AD-IND secure. Putting everything together, we obtain many-AD-IND security of the MIFE in Fig. 10.

Theorem 3 (one-AD-IND security of $\mathcal{MIFE}$). *Suppose the* $\mathcal{D}_k$-*MDDH assumption holds in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$. *Then, the multi-input FE in Fig. 10 is one-AD-IND-secure.*

That is, we show that our multi-input FE is adaptively secure when there is only a single challenge ciphertext.

Game	$\mathbf{c}_i$	$\mathbf{c}_i'$	$\mathbf{c}_i''$	$\{\mathbf{d}_i\}_{i\in[n]}$:	z:	justification/remark	reference
0	$\mathbf{A}\mathbf{s}_i$	$\mathbf{W}_i\mathbf{c}_i + \mathbf{x}_i^\beta$	$\mathbf{V}_i\mathbf{c}_i + \mathbf{z}_i$	$\mathbf{W}_i^\top\mathbf{y}_i + \mathbf{V}_i\mathbf{r}$	$\langle \mathbf{z}_1 + \ldots + \mathbf{z}_n, \mathbf{r}\rangle$	one-AD-IND security game	Definition 7
1	$\mathbf{A}\mathbf{s}_i + \boxed{\mathbf{u}}$	$\mathbf{W}_i\mathbf{c}_i + \mathbf{x}_i^\beta$	$\mathbf{V}_i\mathbf{c}_i + \mathbf{z}_i$	$\mathbf{W}_i^\top\mathbf{y}_i + \mathbf{V}_i\mathbf{r}$	$\langle \mathbf{z}_1 + \ldots + \mathbf{z}_n, \mathbf{r}\rangle$	$\mathcal{D}_k$-MDDH in $\mathbb{G}_1$	Lemma 10
2	$\mathbf{A}\mathbf{s}_i + \mathbf{u}$	$\mathbf{W}_i\mathbf{c}_i + \mathbf{x}_i^\beta$	$\boxed{\mathbf{V}_i\mathbf{c}_i}$	$\mathbf{W}_i^\top\mathbf{y}_i + \mathbf{V}_i\mathbf{r} - \boxed{\mathbf{a}^\perp\langle \mathbf{z}_i, \mathbf{r}\rangle}$	$\langle \mathbf{z}_1 + \ldots + \mathbf{z}_n, \mathbf{r}\rangle$	inf. theoretic	Lemma 11
3	$\mathbf{A}\mathbf{s}_i + \mathbf{u}$	$\mathbf{W}_i\mathbf{c}_i + \mathbf{x}_i^\beta$	$\mathbf{V}_i\mathbf{c}_i$	$\mathbf{W}_i^\top\mathbf{y}_i + \mathbf{V}_i\mathbf{r} - \boxed{\mathbf{a}^\perp\tilde{z}_i}$	$\boxed{\sum_i \tilde{z}_i}$	$\mathcal{D}_k$-MDDH in $\mathbb{G}_2$	Lemma 12
3*	$\mathbf{A}\mathbf{s}_i + \mathbf{u}$	$\mathbf{W}_i\mathbf{c}_i + \mathbf{x}_i^\beta$	$\mathbf{V}_i\mathbf{c}_i$	$\mathbf{W}_i^\top\mathbf{y}_i + \mathbf{V}_i\mathbf{r} - \boxed{\mathbf{a}^\perp\tilde{z}_i}$	$\boxed{\sum_i \tilde{z}_i}$	selective variant	Lemma 13
4*	$\mathbf{A}\mathbf{s}_i + \mathbf{u}$	$\boxed{\mathbf{W}_i\mathbf{c}_i}$	$\mathbf{V}_i\mathbf{c}_i$	$\mathbf{W}_i^\top\mathbf{y}_i + \mathbf{V}_i\mathbf{r} - \mathbf{a}^\perp\tilde{z}_i$	$\sum_i \tilde{z}_i - \boxed{\sum_i \langle \mathbf{x}^\beta, \mathbf{y}_i\rangle}$	inf. theoretic	Lemma 14, Lemma 15

Fig. 11: Sequence of games for the proof of Theorem 3. Here, for any slot $i \in [n]$, $([\mathbf{c}_i]_1, [\mathbf{c}_i']_1, [\mathbf{c}_i'']_1)$ is the challenge ciphertext computed by $\mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$; $[\mathbf{d}_i]_2$ and $[z]_T$ are part of the $\mathsf{sk}_{\mathbf{y}_1, \|\cdots\|\mathbf{y}_n}$ computed by $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)$. We use $\mathbf{u} \leftarrow_\mathrm{R} \mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A})$ and $\mathbf{a}^\perp \leftarrow_\mathrm{R} \mathbb{Z}_q^{k+1}$ such that $\mathbf{A}^\top\mathbf{a}^\perp = \mathbf{0}$ and $\langle \mathbf{u}, \mathbf{a}^\perp\rangle = 1$. In the last column, ad. (resp. sel.) refers to the fact that the game is adaptive (resp. selective). To analyze Game_3, we consider the selective variant of the game: $\mathsf{Game}_{3\star}$, prove using an information-theoretic argument (via $\mathsf{Game}_{4\star}$) that the advantage is 0, and deduce that the advantage in Game_3 is also 0 via complexity leveraging. Note that both $\mathsf{Game}_{3\star}$ and $\mathsf{Game}_{4\star}$ are selective.

Game$_i$, for $i \in \left\{ \boxed{0}, \boxed{1,2,3}, \boxed{3^\star, 4^\star} \right\}$:

$\{\mathbf{x}_i^b\}_{i\in[n]} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F})$

$\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1)\times k}$, $\boxed{\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1} \setminus \mathsf{Span}(\mathbf{A}), \mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1} \text{ s.t. } \mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0} \text{ and } \mathbf{u}^\top \mathbf{a}^\perp = 1}$

For all $i \in [n]$: $\mathbf{W}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m\times(k+1)}$, $\mathbf{V}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times(k+1)}$, $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, $\beta \leftarrow_{\mathrm{R}} \{0,1\}$,

$\mathsf{ct}_i := \mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$

$\mathsf{msk} := \left\{ \mathbf{W}_i, \mathbf{V}_i, \mathbf{z}_i \right\}_{i\in[n]}$, $\mathsf{mpk} := \left([\mathbf{A}]_1, \{[\mathbf{A}^\top \mathbf{W}_i]_1, [\mathbf{A}^\top \mathbf{V}_i]_1\}_{i\in[n]}\right)$

$\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot),\ \boxed{\mathsf{Enc}(\cdot,\cdot,\cdot)}} \left(\mathsf{mpk}, \boxed{\{\mathsf{ct}_i\}_{i\in[n]}}\right)$

Output: 1 if $\beta' = \beta$, 0 otherwise.

$\underline{\mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^1):}$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad 0, 1, \boxed{2,3,3^\star}, \boxed{4^\star}$

$\mathbf{s}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k; [\mathbf{c}_i]_1 := [\mathbf{A}\mathbf{s}_i]_1 + \boxed{[\mathbf{u}]_1}$

$[\mathbf{c}_i']_1 := \mathbf{W}_i[\mathbf{c}_i]_1 + [\mathbf{x}_i^\beta]_1;\ \boxed{\mathbf{W}_i[\mathbf{c}_i]_1}$

$[\mathbf{c}_i'']_1 := \mathbf{V}_i[\mathbf{c}_i]_1 + [\mathbf{z}_i]_1;\ \boxed{[\mathbf{c}_i'']_1 := \mathbf{V}_i[\mathbf{c}_i]_1}$

Return $\mathsf{ct}_i := ([\mathbf{c}_i]_1, [\mathbf{c}_i']_1, [\mathbf{c}_i'']_1)$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n \in (\mathbb{Z}_q^m)^n):}$ $\qquad\qquad\qquad 0,1, \boxed{2}, \boxed{3,3^\star}, \boxed{4^\star}$

$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k; \boxed{\tilde{z}_1, \ldots, \tilde{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p}$

$\mathbf{d}_i := \mathbf{W}_i^\top \mathbf{y}_i + \mathbf{V}_i^\top \mathbf{r} - \boxed{\mathbf{a}^\perp \langle \mathbf{z}_i, \mathbf{r} \rangle} - \boxed{\mathbf{a}^\perp \tilde{z}_i}$

$z := \langle \mathbf{z}_1 + \ldots + \mathbf{z}_n, \mathbf{r} \rangle;\ \boxed{z := \sum_i \tilde{z}_i - \boxed{\sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle}}$

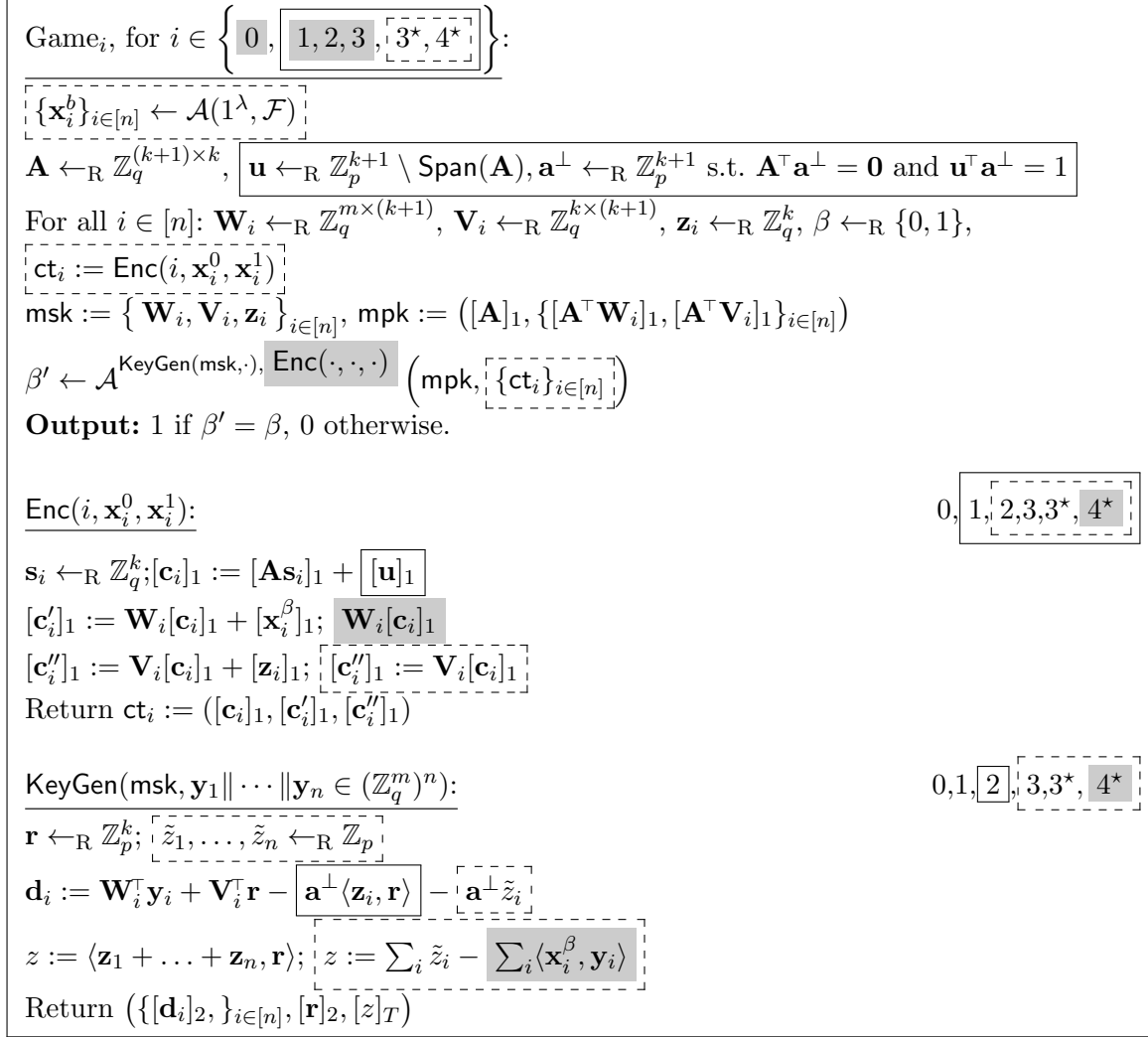Return $\left(\{[\mathbf{d}_i]_2, \}_{i\in[n]}, [\mathbf{r}]_2, [z]_T\right)$

Fig. 12: Games$_i$, for $i \in \{0, \ldots, 3, 3^\star, 4^\star\}$ for the proof of Theorem 3. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame.

Proof (of Theorem 3). We proceed via a series of Game$_i$ for $i \in \{0, \ldots, 3, 3^\star, 4^\star\}$ described in Fig. 12. Let $\mathcal{A}$ be a PPT adversary, and $\lambda \in \mathbb{N}$ be the security parameter. For all $i \in \{0, \ldots, 3, 3^\star, 4^\star\}$, we define $\mathsf{Adv}_i(\mathcal{A}) := \Pr[\mathsf{Game}_i(1^\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and Game$_i$. The transitions are summarized in Fig. 11.

Game$_0$: is the experiment **one-AD-IND**$^{\mathcal{MIFE}}(1^\lambda, \mathcal{A})$ (see Definition 7). In particular, $\mathsf{Adv}_0(\mathcal{A}) = \mathsf{Adv}^{\mathcal{MIFE}, one-AD-IND}(\lambda, \mathcal{A})$.

Game$_1$: we change the distribution of the vectors $[\mathbf{c}_i]_1$ computed by $\mathsf{Enc}(i, \cdot, \cdot)$, for all slots $i \in [n]$, using the $\mathcal{D}_k$-MDDH assumption (cf Lemma 10).

Lemma 10 (Game$_0$ to Game$_1$). *There exists a PPT adversary $\mathcal{B}_1$ such that:*

$$\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_1, \mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}.$$

Proof (of Lemma 10). Here, we switch $\{[\mathbf{A}]_1, [\mathbf{A}\mathbf{s}_i]_1\}_{i \in [n]}$ computed by $\mathsf{Enc}(i, \cdot, \cdot)$ to $\{[\mathbf{A}]_1, [\mathbf{A}\mathbf{s}_i + \boxed{\mathbf{u}}]_1\}_{i \in [n]}$, where $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{(k+1) \times k}$, $\mathbf{u} \leftarrow_R \mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A})$, and for all $i \in [n]$, $\mathbf{s}_i \leftarrow_R \mathbb{Z}_q^k$. This change is justified by the facts that:

1. The distributions: $\{\mathbf{s}_i\}_{i \in [n]}$ and $\{\mathbf{s}_i + \mathbf{s}\}_{i \in [n]}$, where $\mathbf{s} \leftarrow_R \mathbb{Z}_q^k$ and for all $i \in [n]$, $\mathbf{s}_i \leftarrow_R \mathbb{Z}_q^k$, are identically distributed.
2. By the $\mathcal{D}_k$-MDDH assumption, we can switch $([\mathbf{A}]_1, [\mathbf{A}\mathbf{s}]_1)$ to $([\mathbf{A}]_1, [\mathbf{u}]_1)$, where $\mathbf{A} \leftarrow_R \mathcal{D}_k$, $\mathbf{s} \leftarrow_R \mathbb{Z}_q^k$, and $\mathbf{u} \leftarrow_R \mathbb{Z}_q^{k+1}$.
3. The uniform distribution over $\mathbb{Z}_q^{k+1}$ and $\mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A})$ are $\frac{1}{q}$-close, for all $\mathbf{A}$ of rank k.

Combining these three facts, we obtain a PPT adversary $\mathcal{B}_0'$ such that

$$\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_1, \mathcal{B}_0'}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}.$$

Now we describe the adversary $\mathcal{B}_1$. Upon receiving an MDDH challenge $(\mathcal{PG}, [\mathbf{A}]_1, [\mathbf{h}]_1)$, $\mathcal{B}_1$ picks $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_q^{m \times (k+1)}$, $\mathbf{V}_1, \ldots, \mathbf{V}_n \leftarrow_R \mathbb{Z}_q^{k \times (k+1)}$, $\mathbf{z}_1, \ldots, \mathbf{z}_n \leftarrow_R \mathbb{Z}_q^k$, thanks to which it can compute mpk and simulate the oracle $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, as described in Fig. 12. To simulate $\mathsf{Enc}(i, \cdot, \cdot)$, $\mathcal{B}_1$ picks $\mathbf{s}_i \leftarrow_R \mathbb{Z}_p^k$, sets $[\mathbf{c}_i]_1 := [\mathbf{A}]_1 \mathbf{s}_i + [\mathbf{h}]_1$, and computes the rest of the challenge ciphertext from $[\mathbf{c}_i]_1$. Note that when $[\mathbf{h}]_1$ is a real MDDH challenge, this simulates Game$_0$, whereas it simulates Game$_1$ when $[\mathbf{h}]_1$ is uniformly random over $\mathbb{G}_1^{k+1}$ (within $\frac{1}{q}$ statistical distance). $\qquad \square$

Game$_2$: here, for all slots $i \in [n]$, we change the way the vectors $[\mathbf{c}_i'']_1$ and $[\mathbf{d}_i]_2$ are computed, respectively, by $\mathsf{Enc}(i, \cdot, \cdot)$ and $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, using an information theoretic argument. The point is to make it possible to simulate Game$_2$ only knowing $[\mathbf{z}_i]_2$ (and not $[\mathbf{z}_i]_1$), which will be useful later, to use the MDDH assumption on $[\mathbf{z}_i]_2$, in $\mathbb{G}_2$.

Lemma 11 (Game 1 to 2). $\mathsf{Adv}_1(\mathcal{A}) = \mathsf{Adv}_2(\mathcal{A}).$

Proof. We argue that Game 1 and 2 are the same, using the fact that for all $\mathbf{A} \in \mathbb{Z}_q^{(k+1) \times k}$, $\mathbf{u} \in \mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A})$, and all $\mathbf{a}^\perp \in \mathbb{Z}_q^{k+1}$ such that $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$ and $(\mathbf{a}^\perp)^\top \mathbf{u} = 1$, the following distributions are identical:

$$\{\mathbf{V}_i, \mathbf{z}_i\}_{i \in [n]} \text{ and } \{\mathbf{V}_i - \boxed{\mathbf{z}_i(\mathbf{a}^\perp)^\top}, \mathbf{z}_i\}_{i \in [n]},$$

where for all $i \in [n]$, $\mathbf{V}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k \times (k+1)}$, and $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$. This is the case because the matrices $\mathbf{V}_i$ are picked uniformly, independently of the vectors $\mathbf{z}_i$. This way, we obtain

$$\mathbf{d}_i := \mathbf{W}_i^\top \mathbf{y}_i + \left(\mathbf{V}_i^\top - \boxed{\mathbf{a}^\perp \mathbf{z}_i^\top} \right) \mathbf{r} = \mathbf{V}_i^\top \mathbf{r} - \boxed{\mathbf{a}^\perp \mathbf{z}_i^\top \mathbf{r}}$$

and

$$\begin{aligned}
[\mathbf{c}_i'']_1 &:= \left[\left(\mathbf{V}_i - \boxed{\mathbf{z}_i(\mathbf{a}^\perp)^\top} \right)\left(\mathbf{A}\mathbf{s}_i + \mathbf{u} \right) \right]_1 + [\mathbf{z}_i]_1 \\
&= \left[\mathbf{V}_i\left(\mathbf{A}\mathbf{s}_i + \mathbf{u} \right) \right]_1 - \boxed{\left[\mathbf{z}_i(\mathbf{a}^\perp)^\top \mathbf{u} \right]_1} + [\mathbf{z}_i]_1 \\
&= \left[\mathbf{V}_i\left(\mathbf{A}\mathbf{s}_i + \mathbf{u} \right) \right]_1
\end{aligned}$$

where we use the fact that $(\mathbf{a}^\perp)^\top \mathbf{u} = 1$ is the last equality. This corresponds to Game_2. $\qquad\square$

Game$_3$: we use the $\mathcal{D}_k$-MDDH assumption to switch simultaneously for all $i \in [n]$ the values $[\langle \mathbf{z}_i, \mathbf{r} \rangle]_2$ computed by $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, to uniformly random values over $\mathbb{G}_2$. This relies on the fact that it is not necessary to know $[\mathbf{z}_i]_1$ to simulate Game_2 or Game_3.

Lemma 12 (Game 2 to 3). *There exists a PPT adversary $\mathcal{B}_3$ such that:*

$$\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_2, \mathcal{B}_3}^{\mathcal{U}_k\text{-MDDH}}(\lambda) + \frac{1}{q-1}.$$

Proof. Here, we switch $\{[\mathbf{r}]_2, [\langle \mathbf{z}_i, \mathbf{r} \rangle]_2\}_{i \in [n]}$ used by $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ to $\{[\mathbf{r}]_2, [\tilde{z}_i]_2\}_{i \in [n]}$, where for all $i \in [n]$, $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, $\tilde{z}_1, \ldots, \tilde{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$. This is justified by the fact that $[\mathbf{r}^\top \| \langle \mathbf{z}_1, \mathbf{r} \rangle \| \cdots \| \langle \mathbf{z}_n, \mathbf{r} \rangle]_2 \in \mathbb{G}_2^{1 \times (k+n)}$ is identically distributed to $[\mathbf{r}^\top \mathbf{U}^\top]_2$ where $\mathbf{U} \leftarrow_{\mathrm{R}} \mathcal{U}_{k+n,k}$ (wlog. we assume that the upper k rows of $\mathbf{U}$ are full rank), which is indistinguishable from a uniformly random vector over $\mathbb{G}_2^{1 \times (k+n)}$, that is, of the form: $[\mathbf{r} \| \tilde{z}_1 \| \cdots \| \tilde{z}_n]_2$, according to the $\mathcal{U}_{k+n,k}$-MDDH assumption. To do the switch simultaneously for all calls to KeyGen, that is, to switch $\{[\mathbf{r}^j]_2, [\langle \mathbf{z}_i, \mathbf{r}^j \rangle]_2\}_{i \in [n], j \in [Q_0]}$ to $\{[\mathbf{r}^j]_2, [\tilde{z}_i^j]_2\}_{i \in [n], j \in [Q_0]}$, where Q_0 denotes the number of calls to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, and for all $i \in [n]$, $\mathbf{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, $\tilde{z}_1^j, \ldots, \tilde{z}_n^j \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and for all $j \in [Q_0]$, $\mathbf{r}^j \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, we use the Q_0-fold $\mathcal{U}_{k+n,k}$-MDDH assumption. Namely, we build a PPT adversary $\mathcal{B}_3'$ such that $\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_2, \mathcal{B}_3'}^{n\text{-fold }\mathcal{U}_{Q_0,k}\text{-MDDH}}(\lambda)$. This, together with Lemma 1 ($\mathcal{U}_k$-MDDH $\Rightarrow$ n-fold $\mathcal{U}_{Q_0,k}$-MDDH), implies the lemma.

-Simulation of mpk: Upon receiving an n-fold $\mathcal{U}_{Q_0,k}$-MDDH challenge

$$\left(\mathcal{PG}, [\mathbf{U}]_2 \in \mathbb{G}_2^{(k+n) \times k}, \left[\mathbf{h}^1 \| \cdots \| \mathbf{h}^{Q_0} \right]_2 \in \mathbb{G}_2^{(k+n) \times Q_0} \right),$$

$\mathcal{B}_3'$ picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$, samples $\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1) \times k}$, $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1} \setminus \mathsf{Span}(\mathbf{A}), \mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$ s.t. $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$ and $\mathbf{u}^\top \mathbf{a}^\perp = 1$, $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m \times (k+1)}$, $\mathbf{V}_1, \ldots, \mathbf{V}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k \times (k+1)}$, $\mathbf{z}_1, \ldots, \mathbf{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and returns $\mathsf{mpk} := \left([\mathbf{A}]_1, \{[\mathbf{A}^\top \mathbf{W}_i]_1, [\mathbf{A}^\top \mathbf{V}_i]_1\}_{i \in [n]} \right)$ to $\mathcal{A}$.

30

-Simulation of $\mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$**:** $\mathcal{B}_3'$ picks $\mathbf{s}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, computes $[\mathbf{c}_i]_1 := [\mathbf{As}_i]_1 + [\mathbf{u}]_1$, $[\mathbf{c}_i'] := \mathbf{W}_i[\mathbf{c}_i]_1 + [\mathbf{x}_i^\beta]_1$, $[\mathbf{c}_i'']_1 := \mathbf{V}_i[\mathbf{c}_i]_1$, and returns $([\mathbf{c}_i]_1, [\mathbf{c}_i']_1, [\mathbf{c}_i'']_1)$ to $\mathcal{A}$.

-Simulation of $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$**:** On the j'th query $\mathbf{y}_1 \| \cdots \| \mathbf{y}_n$ of $\mathcal{A}$ to KeyGen', $\mathcal{B}_3'$ sets $[\mathbf{r}]_2 := [\overline{\mathbf{h}^j}]_2$, where $\overline{\mathbf{h}^j} \in \mathbb{Z}_q^k$ denotes the k-upper components of $\mathbf{h}^j \in \mathbb{Z}_q^{k+n}$, and for each $i \in [n]$, computes $[\mathbf{d}_i]_2 := [\mathbf{W}_i^\top \mathbf{y}_i]_2 + \mathbf{V}_i^\top [\overline{\mathbf{h}^j}]_2 - \mathbf{a}^\perp [\mathbf{h}_{k+i}^j]_2$, where $\mathbf{h}_{k+i}^j$ denotes the $k + i$'th coordinate of the vector $\mathbf{h}^j \in \mathbb{Z}_p^{k+n}$.

Note that when $\left[\mathbf{h}^1 \| \cdots \| \mathbf{h}^{Q_0}\right]_2$ is a real MDDH challenge, $\mathcal{B}_3'$ simulate Game_2, whereas it simulates Game_3 when $\left[\mathbf{h}^1 \| \cdots \| \mathbf{h}^{Q_0}\right]_2$ is uniformly random over $\mathbb{G}_1^{(k+n) \times Q_0}$. $\qquad \square$

Next, we prove that $\mathsf{Adv}_3(\mathcal{A}) = 0$. To do so, we first consider the selective variant of Game_3, which we call $\mathsf{Game}_{3\star}$ (it is defined in Fig. 12, and the one-SEL-IND security is defined in Definition 5). We prove that for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{3\star}(\mathcal{B}) = 0$, using a hybrid $\mathsf{Game}_{4\star}$. That is, we prove that $\mathsf{Game}_{3\star}$ and $\mathsf{Game}_{4\star}$ are perfectly indistinguishable in Lemma 14, and we prove that for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{4\star}(\mathcal{B}) = 0$, in Lemma 15. Finally, we use complexity leveraging to obtain perfect security of the adaptive Game_3, in Lemma 13.

$\mathsf{Game}_{3\star}$**:** is the selective variant of Game_3, that is, any adversary playing this game has to commit to its challenge queries $\{\mathbf{x}_i^0, \mathbf{x}_i^1\}_{i \in [n]}$ beforehand. In the following lemma, we build an adversary $\mathcal{B}_{3\star}$ against $\mathsf{Game}_{3\star}$, such that $\mathsf{Adv}_3(\mathcal{A}) \leq q^{2nm} \cdot \mathsf{Adv}_{3\star}(\mathcal{B}_{3\star})$. Note that this exponential security loss will not affect the advantage of $\mathcal{A}$, since we will prove that $\mathsf{Adv}_{3\star}(\mathcal{B}_{3\star}) = 0$.

Lemma 13 (Game_3 to $\mathsf{Game}_{3\star}$). *We build a PPT adversary $\mathcal{B}_{3\star}$ such that $\mathsf{Adv}_3(\mathcal{A}) \leq q^{2nm} \cdot \mathsf{Adv}_{3\star}(\mathcal{B}_{3\star})$.*

Proof. $\mathsf{Game}_{3\star}$ is exactly the same than Game_3 except that the former is selective: any adversary playing $\mathsf{Game}_{3\star}$ has to commits to its challenge $\{\mathbf{x}_i^b\}_{i \in [n], b \in \{0,1\}}$ beforehand. We use complexity leveraging to build an adversary in the selective $\mathsf{Game}_{3\star}$ from the adversary $\mathcal{A}$ in the adaptive Game_3 (these game are described in Fig. 12).

The adversary $\mathcal{B}_{3\star}$ against $\mathsf{Game}_{3\star}$ first guesses the future queries $\{\mathbf{x}_i^b\}_{i \in [n], b \in \{0,1\}}$ of $\mathcal{A}$, aborts (and loses the game) if the guess is unsuccessful; otherwise, it behaves exactly as $\mathcal{A}$ (it does the same queries to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$). It is clear that the advantage of $\mathcal{B}_{3\star}$ in $\mathsf{Game}_{3\star}$ is at least $q^{-2mn} \cdot \mathsf{Adv}_3(\mathcal{A})$. $\qquad \square$

$\mathsf{Game}_{4\star}$**:** is like $\mathsf{Game}_{3\star}$, except that for all slots $i \in [n]$, $\mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^0)$ computes $[\mathbf{c}_i']_1 := \mathbf{W}_i[\mathbf{c}_i]_1$, and $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n)$ computes $z := \sum_i \tilde{z}_i - \sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle$. Note We show in Lemma 14 that $\mathsf{Game}_{3\star}$ and $\mathsf{Game}_{4\star}$ are perfectly indistinguishable, using an information theoretic argument that crucially relies on the fact that $\mathsf{Game}_{3\star}$ and $\mathsf{Game}_{4\star}$ are *selective*. Then, in Lemma 15, we show that $\mathsf{Game}_{4\star}$ is independent of β, that is, for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{4\star}(\mathcal{B}) = 0$. This relies on the fact that in $\mathsf{Game}_{4\star}$, β only shows up in the sum $\sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i \rangle$, and the restrictions imposed by the ideal functionality.

31

Lemma 14 (Game$_{3\star}$ to Game$_{4\star}$). *For all adversaries $\mathcal{B}$, $\mathsf{Adv}_{3\star}(\mathcal{B}) = \mathsf{Adv}_{4\star}(\mathcal{B})$.*

Proof. This transition is similar to the transition between Game$_2$ and Game$_3$ in the one-SEL-IND security proof of $\mathcal{MIFE}$ (Lemma 6).

Namely, we use the fact that for all $\mathbf{y}_1\|\cdots\|\mathbf{y}_n \in (\mathbb{Z}_q^m)^n$, for all $\{\mathbf{x}_i^b \in \mathbb{Z}_q^m\}_{i\in[n],b\in\{0,1\}}$, all $\beta \in \{0,1\}$, the following are identically distributed: $\{\tilde{z}_i\}_{i\in[n]}$ and $\{\tilde{z}_i - \boxed{\langle \mathbf{x}_i^\beta, \mathbf{y}_i\rangle}\}_{i\in[n]}$, where $\tilde{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ for all $i \in [n]$.

For each query $\mathbf{y}_1\|\cdots\|\mathbf{y}_n$, $\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)$ picks values $\tilde{z}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ for $i \in [n]$ that are *independent* of $\mathbf{y}_1\|\cdots\|\mathbf{y}_n$ and the challenge $\{\mathbf{x}_i^b \in \mathbb{Z}_q^m\}_{i\in[n],b\in\{0,1\}}$ (note that here we crucially rely on the fact the Game$_{3\star}$ and Game$_{4\star}$ are *selective*), therefore, using the previous fact, we can switch $\tilde{z}_i$ to $\tilde{z}_i - \boxed{\langle \mathbf{x}_i^\beta, \mathbf{y}_i\rangle}$ without changing the distribution of the game. This way, $\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n)$ computes $z := \sum_i \tilde{z}_i - \sum_i \boxed{\langle \mathbf{x}_i^\beta, \mathbf{y}_i\rangle}$, exactly as in Game$_{4\star}$.

Similarly, using the fact that for all $\mathbf{x}_i^\beta \in \mathbb{Z}_q^m$, and $\mathbf{a}^\perp \in \mathbb{Z}_q^{k+1}$: $\mathbf{W}_i$, and $\mathbf{W}_i - \boxed{\mathbf{x}_i^\beta(\mathbf{a}^\perp)^\top}$ are identically distributed, where $\mathbf{W}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m\times(k+1)}$; and the fact that in both Game$_{3\star}$ and Game$_{4\star}$, the challenge $\{\mathbf{x}_i^b\}_{i\in[n],b\in\{0,1\}}$ is independent of $\mathbf{W}_i$ (again, this is the case because both games are selective), we can switch, for all $i \in [n]$, the vector $\mathbf{c}_i'$, computed by $\mathsf{Enc}(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$ to: $\mathbf{c}_i' := \mathbf{W}_i[\mathbf{c}_i]$, as in Game$_{4\star}$. $\qquad\square$

Lemma 15 (Game$_{4\star}$). *For all adversaries $\mathcal{B}$, $\mathsf{Adv}_{4\star}(\mathcal{B}) = 0$.*

Proof. We use the fact that for all $i \in [n]$, the query $(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$ to Enc' (recall that there can be at most one query per slot $i \in [n]$), and for all queries $\mathbf{y}_1\|\cdots\|\mathbf{y}_n$ to KeyGen', we have:

$$\sum_{i=1}^n \langle \mathbf{x}_i^0, \mathbf{y}_i\rangle = \sum_{i=1}^n \langle \mathbf{x}_i^1, \mathbf{y}_i\rangle.$$

Therefore, for each call to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, the value z, which is of the form $z := \sum_i \tilde{z}_i - \sum_i \langle \mathbf{x}_i^\beta, \mathbf{y}_i\rangle$, is independent of β. Since the challenge ciphertext and the public key are also independent of β, we have $\mathsf{Adv}_{4\star}(\mathcal{B}) = 0$. $\qquad\square$

Summing up, we have shown in the previous lemmas that for all PPT adversaries $\mathcal{A}$ and $\lambda \in \mathbb{N}$ we have the following.

- In Lemma 10, we build a PPT adversary $\mathcal{B}_1$ such that $\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_1,\mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}$.
- In Lemma 11, we show $\mathsf{Adv}_1(\mathcal{A}) = \mathsf{Adv}_2(\mathcal{A})$.
- In Lemma 12, there exists a PPT adversary $\mathcal{B}_3$ such that the $\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_2,\mathcal{B}_3}^{\mathcal{U}_k\text{-MDDH}}(\lambda) + \frac{1}{q-1}$.
- In Lemma 13, we build a PPT adversary $\mathcal{B}_{3\star}$ such that $\mathsf{Adv}_3(\mathcal{A}) \leq q^{2nm} \cdot \mathsf{Adv}_{3\star}(\mathcal{B}_{3\star})$, using complexity leveraging.
- In Lemma 14, we show that for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{3\star}(\mathcal{B}) = \mathsf{Adv}_{4\star}(\mathcal{B})$.
- In Lemma 15, we show that for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{4\star}(\mathcal{B}) = 0$, using the restrictions on the queries to oracles $\mathsf{Enc}(\cdot,\cdot,\cdot)$ and $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ imposed by the definition of the security game.

game	$\mathsf{Enc}'(i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1})$:	justification/remark
0	$\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \mathbf{x}_i^{1,0})$	many-AD-IND security game
1	$\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \boxed{\mathbf{x}_i^{1,1}})$	one-AD-IND security of $\mathcal{MIFE}$
$1.\ell$	$\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1})$ for $i \leq \ell$ $\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}} + \mathbf{x}_i^{1,1})$ for $i > \ell$	many-AD-IND security of $\mathcal{FE}$
2	$\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1}})$	game 2 is game $1.n$

Fig. 13: Sequence of games for the proof of Theorem 4.

Combining everything together, we obtain:

$$\mathsf{Adv}_0(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}_1, \mathcal{B}_1}^{\mathcal{U}_k\text{-MDDH}}(\lambda) + \mathsf{Adv}_{\mathbb{G}_2, \mathcal{B}_3}^{\mathcal{U}_k\text{-MDDH}}(\lambda) + \frac{2}{q-1}.$$

Using the $\mathcal{U}_k$-MDDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$ (which is implied by k-Lin, see Lemma 2), this means $\mathsf{Adv}_0(\mathcal{A})$ is negligible in λ. $\qquad\square$

Remark 5 (On adaptive security). To achieve *adaptive* security, we split the selective, computational argument used for the proof of Theorem 1, in two steps: first, we use an adaptive, computational argument, that does not involve the challenges $\{\mathbf{x}_i^b\}_{i \in [n], b\{0,1\}}$ (this corresponds to the transition from Game_0 to Game_3). Then, we prove security of Game_3, using a selective argument, which involves the challenges $\{\mathbf{x}_i^b\}_{i \in [n], b\{0,1\}}$, but relies on perfect indistinguishability. That is, we prove that Game_3 is perfectly secure, by first proving the perfect security of its selective variant, $\mathsf{Game}_{3\star}$ (this involves an extra hybrid game $\mathsf{Game}_{4\star}$), and finally using complexity leveraging to obtain security of the adaptive Game_3. The proof of Theorem 1 essentially does the two step at once, which prevents using complexity leveraging.

Theorem 4 (many-AD-IND security of $\mathcal{MIFE}$). *Suppose the single-input FE* (Setup, Enc, KeyGen, Dec) *is many-AD-IND-secure and the multi-input FE* (Setup', Enc', KeyGen', Dec') *is one-AD-IND-secure. Then, the multi-input FE* (Setup', Enc', KeyGen', Dec') *is many-AD-IND-secure.*

That is, we show that our multi-input FE is adaptively secure in the setting with multiple challenge ciphertexts (and since our multi-input FE is a private key scheme, one-AD-IND security does not immediately imply many-AD-IND security).

The proof of this theorem is essentially the same of that of Theorem 2. Indeed, we follow the same proof strategy (see the proof overview of Theorem 2): first switch the first challenge ciphertexts ct_i^1 for all slots $i \in [n]$ simultaneously, using the one-AD-IND security the multi-input FE; then, using a hybrid argument across slots $i \in [n]$, switch all the remaining ciphertexts ct_i^j for $j \in [Q_i]$, using the many-AD-IND of the single-input FE. These arguments scale well to the adaptive setting: since we start out with adaptive security, we naturally obtain a many-AD-IND secure MIFE. We give a high-level illustration of the transitions for the proof of this theorem in Fig. 13.

$$
\boxed{
\begin{array}{l}
\underline{\text{Game}_0(1^\lambda, \mathcal{A}):} \\
\quad (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B}) \\
\quad \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk}, \cdot), \mathsf{Enc}'(\cdot, \cdot, \cdot)}(\mathsf{mpk}) \\
\quad \text{return } \beta' \\
\underline{\mathsf{Enc}'(i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}):} \\
\quad \text{return } \mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \mathbf{x}_i^{1,0}) \\
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\
\quad \text{return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
\quad
\begin{array}{l}
\underline{\text{Game}_1(1^\lambda, \mathcal{A}):} \\
\quad (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B}) \\
\quad \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk}, \cdot), \mathsf{Enc}'(\cdot, \cdot, \cdot)}(\mathsf{mpk}) \\
\quad \text{return } \beta' \\
\underline{\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}):} \\
\quad \text{return } \mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \boxed{\mathbf{x}_i^{1,1}}) \\
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\
\quad \text{return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
}
$$

$$
\boxed{
\begin{array}{l}
\underline{\text{Game}_{1.\ell}(1^\lambda, \mathcal{A}):} \\
\quad (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B}) \\
\quad \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk}, \cdot), \mathsf{Enc}'(\cdot, \cdot, \cdot)}(\mathsf{mpk}) \\
\quad \text{return } \beta' \\
\underline{\mathsf{Enc}'(i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}):} \\
\quad \text{If } i \leq \ell: \\
\quad\quad \text{return } \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1}) \\
\quad \text{If } i > \ell: \\
\quad\quad \text{return } \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}} + \mathbf{x}_i^{1,1}) \\
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\
\quad \text{return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
\quad
\begin{array}{l}
\\
\underline{\text{Game}_2(1^\lambda, \mathcal{A}):} \\
\quad (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}'(1^\lambda, \mathcal{F}_n^{m,B}) \\
\quad \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}'(\mathsf{msk}, \cdot), \mathsf{Enc}'(\cdot, \cdot, \cdot)}(\mathsf{mpk}) \\
\quad \text{return } \beta' \\
\underline{\mathsf{Enc}'(i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}):} \\
\quad \text{return } \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1}) \\
\underline{\mathsf{KeyGen}'(\mathsf{msk}, \mathbf{y}_1 \| \cdots \| \mathbf{y}_n):} \\
\quad \text{return } \mathsf{sk}_{\mathbf{y}_1 \| \cdots \| \mathbf{y}_n}
\end{array}
}
$$

Fig. 14: $\text{Game}_{0,1,2}$, and $\text{Game}_{1.\ell}$ where $\ell \in \{0, \dots, n\}$ for the proof of Theorem 4. The description of Setup' and KeyGen' are omitted since their simulations do not change throughout the proof.

Proof (of Theorem 4).

We proceed via a series of Game_i for $i \in \{0, 1, 2\}$, and $\text{Game}_{1.\ell}$ for $\ell \in \{0, \dots, n\}$, described in Fig. 14. The transitions are summarized in Fig. 13. Let $\mathcal{A}$ be a PPT adversary and $\lambda \in \mathbb{N}$ be the security parameter. For any adversary $\mathcal{A}$ and any security parameter , For all $i \in \{0, 1, 2, 1.0, \dots, 1.n\}$, we define $\mathsf{Adv}_i(\mathcal{A}) := \Pr[\text{Game}_i(1^\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and Game_i.

Game$_0$: is the experiment **many-AD-IND**$_0^{\mathcal{MIFE}}(1^\lambda, \mathcal{A})$ (see Definition 7).

Game$_1$: we change the challenge ciphertexts to $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \boxed{\mathbf{x}_i^{1,1}})$, for all slots $i \in [n]$, and all $j \in [Q_i]$, using the one-AD-IND security of $\mathcal{MIFE}$ (cf Lemma 16).

Lemma 16 (Game$_0$ to Game$_1$). *There exists a PPT adversary $\mathcal{B}_1$ such that*

$$
|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{MIFE}, one\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_1).
$$

Proof. In Game$_1$, which is described in Fig. 14, we replace $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{j,0}) = \mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{1,0}} +$
$(\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}))$ with $\mathsf{Enc}(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{1,1}} + (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}))$ for all $i \in [n], j \in [Q_i]$. This is justified using the
following properties:

- one-AD-IND security of $(\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$;
- given $\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}$, we can maul an encryption of $\mathbf{x}_i^{1,\beta}$ under Enc' (corresponding to challenge
  ciphertext in slot i in **one-AD-IND**$_\beta^{\mathcal{MIFE}}$) into that of $\mathbf{x}_i^{1,\beta} + (\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})$ (corresponding to
  challenge ciphertexts in slots i in Game$_\beta$).

We build $\mathcal{B}_1$, a PPT adversary such that for all $\beta \in \{0, 1\}$, when $\mathcal{B}_1$ interacts with the experiment
one-AD-IND$_\beta^{\mathcal{MIFE}}$ (see Definition 7), it simulates the Game$_\beta$ to $\mathcal{A}$.

-Simulation of mpk: $\mathcal{B}_1$ first gets mpk, that it sends to $\mathcal{A}$.

-Simulation of $\mathsf{Enc}'(\cdot, \cdot, \cdot)$: Then, given $\mathcal{A}$'s first query $(i, \mathbf{x}_i^{1,0}, \mathbf{x}_i^{1,1})$ for slot $i \in [n]$ to $\mathsf{Enc}'(\cdot, \cdot, \cdot)$,
$\mathcal{B}_1$ computes the first challenge ciphertext ct_i^1 as $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{1,\beta})$, where the oracle $\mathsf{Enc}'(\cdot, \cdot, \cdot)$ of
the experiment **one-AD-IND**$_\beta^{\mathcal{MIFE}}$.

Next, $\mathcal{B}_1$ uses the fact that the single-input inner-product scheme is public-key and linearly ho-
momorphic to generate all the remaining ciphertexts ct_i^j for $i \in [n], j \in \{2, \ldots, Q_i\}$ by combining
$\mathsf{ct}_i^1 = \mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{1,\beta}) = \mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{1,\beta} \| \mathbf{z}_i)$ with $\mathsf{Enc}(\mathsf{msk}_i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} \| \mathbf{0})$ (which can be done using
mpk_i). In particular, each ct_i^j will correspond to $\mathsf{Enc}'(\mathsf{msk}, i, \mathbf{x}_i^{1,\beta} + \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})$, which matches the
challenge ciphertexts in Game$_\beta$.

-Simulation of $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$: $\mathcal{B}_1$ uses its $\mathsf{KeyGen}'(\mathsf{msk}, \cdot)$ oracle to simulate the corresponding
queries by $\mathcal{A}$.

Finally, $\mathcal{B}_1$ outputs 1 if $\mathcal{A}$ outputs 1, 0 otherwise. It is clear that for all $\beta \in \{0, 1\}$, when $\mathcal{B}_1$ interacts
with **one-AD-IND**$_\beta^{\mathcal{MIFE}}(1^\lambda, \mathcal{B}_1)$, it simulates game β to $\mathcal{A}$. Therefore,

$$
\begin{aligned}
&\mathsf{Adv}^{\mathcal{MIFE}, one\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_1) \\
&= \left| \Pr\left[\textbf{one-AD-IND}_0^{\mathcal{MIFE}}(1^\lambda, \mathcal{B}_1) = 1 \right] - \Pr\left[\textbf{one-AD-IND}_1^{\mathcal{MIFE}}(1^\lambda, \mathcal{B}_1) = 1 \right] \right| \\
&= |\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})|
\end{aligned}
$$

$\square$

Game$_2$: we change the challenge ciphertexts to $\mathsf{Enc}'(\mathsf{msk}, i, \boxed{\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}} + \mathbf{x}_i^{1,1})$, for all slots $i \in [n]$,
and all $j \in [Q_i]$, using the many-AD-IND security of $\mathcal{FE}$, via a hybrid argument across all slots
$i \in [n]$ (cf Lemma 17).

Lemma 17 (Game$_1$ to Game$_2$). *There exists a PPT adversary $\mathcal{B}_2$ such that* $|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq n \cdot \mathsf{Adv}^{\mathcal{FE}, many\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_2)$.

Proof. In Game_2, which is described in Fig. 14, we replace $\text{Enc}(\text{msk}_i, \mathbf{x}_i^{1,1} + \boxed{(\mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})}\|\mathbf{z}_i)$ with $\text{Enc}(\text{msk}_i, \mathbf{x}_i^{1,1} + \boxed{(\mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1})}\|\mathbf{z}_i) = \text{Enc}(\text{msk}_i, \mathbf{x}_i^{j,1}\|\mathbf{z}_i)$, for all $i \in [n], j \in [Q_i]$. This is justified using the following properties:

- many-AD-IND security of $(\text{Setup}, \text{Enc}, \text{KeyGen}, \text{Dec})$;
- given $\mathbf{x}_i^{1,1}$ and $\mathbf{z}_i$, we can maul an encryption of $(\mathbf{x}_i^{j,\beta} - \mathbf{x}_i^{1,\beta})\|\mathbf{0}$ under Enc (corresponding to challenge ciphertexts in the many-AD-IND security game) into that of $\mathbf{x}_i^{1,1} + \boxed{(\mathbf{x}_i^{j,\beta} - \mathbf{x}_i^{1,\beta})}\|\mathbf{z}_i$ (corresponding to challenge ciphertexts in $\text{Game}_{\beta+1}$);
- the fact that for each key query $\mathbf{y}_1\|\ldots\|\mathbf{y}_n$ and all $\mathbf{r}, \mathbf{z}$, we have

$$\langle \mathbf{x}_i^{1,1} + \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}\|\mathbf{z}, \mathbf{y}_i\|\mathbf{r}\rangle = \langle \mathbf{x}_i^{1,1} + \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}\|\mathbf{z}, \mathbf{y}_i\|\mathbf{r}\rangle.$$

The latter is equivalent to $\langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i\rangle = \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i\rangle$, which follows from the restriction in Section 3.2.

In more details, we use a hybrid argument, which involves hybrid $\text{Game}_{1.\ell}$ for $\ell \in \{0, \ldots, n\}$, defined in Fig. 14.

Notice that Game_1 and Game_2 are identically distributed than $\text{Game}_{1.0}$ and $\text{Game}_{1.n}$, respectively. For all $\ell \in [n]$, we builds a PPT adversary $\mathcal{B}_{1.\ell}$ such that for all $\beta \in \{0, 1\}$, when $\mathcal{B}_{1.\ell}$ interacts with the experiment $\textbf{many-AD-IND}_\beta^{\mathcal{FE}}$ (see Definition 7), it simulates $\text{Game}_{1.\ell-1+\beta}$ to $\mathcal{A}$.

-Simulation of mpk: First, $\mathcal{B}_{1.\ell}$ receives mpk_ℓ, picks $\mathbf{z}_i \leftarrow_\text{R} \mathbb{Z}_q^k$ for all $i \in [n]$, samples $(\text{mpk}_i, \text{msk}_i) \leftarrow_\text{R} \text{Setup}(1^\lambda, \mathcal{F}_1^{m+k,B})$ for $i \in [n], i \neq \ell$ and returns $\text{mpk} := (\text{mpk}_1, \ldots, \text{mpk}_n)$ to $\mathcal{A}$.

-Simulation of $\text{Enc}'(\cdot, \cdot, \cdot)$: $\mathcal{B}_{1.\ell}$ answers each query $(i, \mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1})$ to Enc' with $i < \ell$ by $\text{Enc}(\text{msk}_i, \mathbf{x}_i^{j,1}\|\mathbf{z}_i)$, and each query with $i > \ell$ by $\text{Enc}(\text{msk}_i, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} + \mathbf{x}_i^{1,1}\|\mathbf{z}_i)$ (note that $\mathcal{B}_{1.\ell}$ can do so since it knows msk_i and $\mathbf{z}_i$ for all $i \neq \ell$). For all queries of the form $(\ell, \mathbf{x}_\ell^{j,0}, \mathbf{x}_\ell^{j,1})$ to Enc', $\mathcal{B}_{1.\ell}$ queries its oracle $\text{Enc}^\beta(\cdot, \cdot, \cdot)$ (from experiment $\textbf{many-AD-IND}_\beta^{\mathcal{FE}}$) on input $(\mathbf{x}_\ell^{j,0}\|\mathbf{z}_i, \mathbf{x}_\ell^{j,1}\|\mathbf{z}_i)$, to obtain $\text{Enc}(\text{msk}_\ell, \mathbf{x}_\ell^{j,\beta}\|\mathbf{z}_\ell)$. $\mathcal{B}_{1.\ell}$ uses the latter together with mpk_ℓ and $(-\mathbf{x}_\ell^{1,0} + \mathbf{x}_\ell^{1,1})\|\mathbf{0}$ to generate $\text{Enc}(\text{msk}_\ell, \mathbf{x}_\ell^{j,\beta} - \mathbf{x}_\ell^{1,0} + \mathbf{x}_\ell^{1,1}\|\mathbf{z}_\ell)$, which it sends to $\mathcal{A}$. This corresponds to ciphertexts in $\text{Game}_{1.\ell-1+\beta}$. This is possible since the single-input inner-product scheme is public-key and linearly homomorphic.

-Simulation of $\text{KeyGen}'(\text{msk}, \cdot)$: For each query $\mathbf{y}_1\|\ldots\|\mathbf{y}_n$ that $\mathcal{A}$ makes to $\text{KeyGen}'(\text{msk}, \cdot)$, $\mathcal{B}_{1.\ell}$ picks $\mathbf{r} \leftarrow_\text{R} \mathbb{Z}_q^k$, and it computes $\mathbf{d}_i \leftarrow \text{KeyGen}(\text{msk}_i, \mathbf{y}_i\|\mathbf{r})$ for $i \in [n], i \neq \ell$. To compute $\mathbf{d}_\ell$, it queries the oracle $\text{KeyGen}(\text{msk}, \cdot)$ on input $\mathbf{y}_\ell\|\mathbf{r}$. It computes $z = \langle \mathbf{z}_1\|\ldots\|\mathbf{z}_n, \mathbf{r}\rangle$ and sends $\text{sk}_{\mathbf{y}_1\|\cdots\|\mathbf{y}_n} := (\{[\mathbf{d}_i]_2\}_{i\in[n]}, [\mathbf{r}]_2, [z]_T)$ to $\mathcal{A}$.

36

Finally, $\mathcal{B}_{1.\ell}$ outputs 1 if $\mathcal{A}$ outputs 1, 0 otherwise. It is clear that for all $\beta \in \{0,1\}$, when $\mathcal{B}_{1.\ell}$ interacts with **many-AD-IND**$_\beta^{\mathcal{MIFE}}$, it simulates Game$_{1.\ell-1+\beta}$ to $\mathcal{A}$. Therefore,

$$\mathsf{Adv}^{\mathcal{MIFE},many\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_{1.\ell})$$
$$= \left| \Pr\left[\textbf{many-AD-IND}_0^{\mathcal{FE}}(1^\lambda, \mathcal{B}_{1.\ell}) = 1 \right] - \Pr\left[\textbf{many-AD-IND}_1^{\mathcal{FE}}(1^\lambda \mathcal{B}_{1.\ell}) = 1 \right] \right|$$
$$= |\mathsf{Adv}_{1.\ell-1}(\mathcal{A}) - \mathsf{Adv}_{1.\ell}(\mathcal{A})|$$

Summing up for all $\ell \in [n]$, we obtain the lemma. $\qquad\square$

Summing up, we prove that for all PPT adversaries $\mathcal{A}$, and all security parameter $\lambda \in \mathbb{N}$, the following is true.

- In Lemma 16, we show that there exists a PPT adversary $\mathcal{B}_1$ such that $|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{MIFE},one\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_1)$.
- In Lemma 17, we show that there exists a PPT adversary $\mathcal{B}_2$ such that $|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq n \cdot \mathsf{Adv}^{\mathcal{FE},many\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_2)$.

Putting everything together, we obtain:

$$|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{MIFE},one\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_1) + n \cdot \mathsf{Adv}^{\mathcal{FE},many\text{-}AD\text{-}IND}(\lambda, \mathcal{B}_2).$$

By Definition 7, $|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| = \mathsf{Adv}^{\mathcal{MIFE},many\text{-}AD\text{-}IND}(\lambda, \mathcal{A})$. Therefore, by the one-AD-IND security of $(\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Enc}')$ and the many-AD-IND security of $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc})$, $\mathsf{Adv}^{\mathcal{MIFE},many\text{-}AD\text{-}IND}(\lambda, \mathcal{A})$ is negligible in λ. $\qquad\square$

References

[1] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 733–751. Springer, Heidelberg, Mar. / Apr. 2015. (Pages 1, 2, and 4.)

[2] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, Report 2016/011, 2016. http://eprint.iacr.org/2016/011. (Pages 1, 4, and 39.)

[3] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 500–518. Springer, Heidelberg, Aug. 2013. (Pages 3, 9, and 46.)

[4] S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 333–362. Springer, Heidelberg, Aug. 2016. (Pages 1, 4, and 39.)

[5] P. Ananth and A. Jain. Indistinguishability obfuscation from compact functional encryption. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Heidelberg, Aug. 2015. (Pages 1, 5, and 6.)

[6] S. Badrinarayanan, D. Gupta, A. Jain, and A. Sahai. Multi-input functional encryption for unbounded arity functions. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 27–51. Springer, Heidelberg, Nov. / Dec. 2015. (Page 1.)

[7] A. Bishop, A. Jain, and L. Kowalczyk. Function-hiding inner product encryption. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 470–491. Springer, Heidelberg, Nov. / Dec. 2015. (Pages 1, 3, and 6.)

[8] N. Bitansky and V. Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In V. Guruswami, editor, *56th FOCS*, pages 171–190. IEEE Computer Society Press, Oct. 2015. (Page 5.)

[9] O. Blazy, E. Kiltz, and J. Pan. (Hierarchical) identity-based encryption from affine message authentication. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Heidelberg, Aug. 2014. (Page 5.)

[10] D. Boneh, K. Lewi, M. Raykova, A. Sahai, M. Zhandry, and J. Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 563–594. Springer, Heidelberg, Apr. 2015. (Pages 1, 2, 5, and 6.)

[11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, Mar. 2011. (Pages 1, 3, 4, and 46.)

[12] Z. Brakerski, I. Komargodski, and G. Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 852–880. Springer, Heidelberg, May 2016. (Pages 1, 5, and 6.)

[13] P. Datta, R. Dutta, and S. Mukhopadhyay. Functional encryption for inner product with full function privacy. In C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 164–195. Springer, Heidelberg, Mar. 2016. (Pages 1 and 6.)

[14] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, Aug. 2013. (Pages 7 and 8.)

[15] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013. (Page 1.)

[16] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, Oct. 2013. (Pages 1 and 6.)

[17] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Functional encryption without obfuscation. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 480–511. Springer, Heidelberg, Jan. 2016. (Page 6.)

[18] R. Gay, D. Hofheinz, E. Kiltz, and H. Wee. Tightly CCA-secure encryption without pairings. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016. (Page 8.)

[19] S. Goldwasser, S. D. Gordon, V. Goyal, A. Jain, J. Katz, F.-H. Liu, A. Sahai, E. Shi, and H.-S. Zhou. Multi-input functional encryption. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 578–602. Springer, Heidelberg, May 2014. (Pages 1, 6, and 8.)

[20] S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 555–564. ACM Press, June 2013. (Page 1.)

[21] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, Aug. 2012. (Page 1.)

[22] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu. Function-hiding inner product encryption is practical. Cryptology ePrint Archive, Report 2016/440, 2016. http://eprint.iacr.org/2016/440. (Pages 2 and 4.)

[23] K. Lee and D. H. Lee. Two-input functional encryption for inner products from bilinear maps. Cryptology ePrint Archive, Report 2016/432, 2016. http://eprint.iacr.org/2016/432. (Pages 2 and 4.)

[24] A. Sahai and H. Seyalioglu. Worry-free encryption: functional encryption with public keys. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *ACM CCS 10*, pages 463–472. ACM Press, Oct. 2010. (Page 1.)

[25] A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005. (Page 1.)

[26] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, Aug. 2009. (Page 20.)

[27] H. Wee. Dual system encryption via predicate encodings. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, Feb. 2014. (Page 5.)

[28] H. Wee. New techniques for attribute-hiding in prime-order bilinear groups. In preparation, 2016. (Pages 4 and 39.)

A One-SEL-SIM, Many-AD-IND Secure Scheme for Single-Input Inner Products

In Fig. 15, we describe the scheme for Single-Input Inner Products from [28], which is essentially the same as those in [4, 2], extended explicitly to the $\mathcal{D}_k$-MDDH assumption. For completeness, we recall the proof of one-SEL-SIM-security from [28] in Theorem 5, as we will rely on some of the proof techniques later. We also prove that the scheme is many-AD-IND secure in Theorem 6. Moreover, note that the scheme is public key, linearly homomorphic, and satisfies additional requirements for the construction in Fig. 3.

$$
\begin{array}{l|l}
\underline{\mathsf{Setup}(\mathbb{G}, \mathcal{F}_1^{m,B})\text{:}} & \underline{\mathsf{Enc}(\mathsf{msk}, \mathbf{x} \in \mathbb{Z}_q^m)\text{:}} \\[4pt]
\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k,\ \mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m \times (k+1)} & \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k; \\
\mathsf{mpk} := ([\mathbf{A}], [\mathbf{WA}]), \mathsf{msk} := (\mathbf{W}, \mathbf{A}); & \text{return } ([\mathbf{c}], [\mathbf{c}']) := ([\mathbf{Ar}], [\mathbf{x} + \mathbf{WAr}]) \\
\text{return } (\mathsf{mpk}, \mathsf{msk}) & \\[10pt]
\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y} \in \mathbb{Z}_q^m)\text{:}} & \underline{\mathsf{Dec}(\mathsf{sk_y}, \mathbf{y}, ([\mathbf{c}], [\mathbf{c}']))\text{:}} \\[4pt]
\text{return } \mathsf{sk_y} := \mathbf{W}^\top \mathbf{y} \in \mathbb{Z}_q^{k+1} & \text{return discrete log of } [\mathbf{c}'^\top \mathbf{y} - \mathbf{c}^\top \mathsf{sk_y}]
\end{array}
$$

Fig. 15: A one-SEL-SIM scheme for single-input inner product $\mathcal{F}_1^{m,B}$ [28].

Theorem 5 (one-SEL-SIM security of $\mathcal{FE}$). [28] *If the $\mathcal{D}_k$-MDDH assumption holds in $\mathbb{G}$, then the single-input FE in Fig. 15 is one-SEL-SIM secure (see Definition 6).*

$$
\begin{array}{l}
\underline{\widetilde{\mathsf{Setup}}(\mathbb{G})\text{:}} \\[4pt]
\quad \mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1) \times k}, \widetilde{\mathbf{W}} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m \times (k+1)}, \mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A}); \\
\quad \text{compute } \mathbf{a}^\perp \in \mathbb{Z}_q^{k+1} \setminus \{\mathbf{0}\} \text{ s.t. } \mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0} \\
\quad \widetilde{\mathsf{mpk}} := ([\mathbf{A}], [\widetilde{\mathbf{W}}\mathbf{A}]), \widetilde{\mathsf{msk}} := (\mathbf{a}^\perp, \widetilde{\mathbf{W}}, \mathbf{c}); \\
\quad \text{return } (\widetilde{\mathsf{mpk}}, \widetilde{\mathsf{msk}}) \\[10pt]
\underline{\widetilde{\mathsf{KeyGen}}(\widetilde{\mathsf{msk}}, \mathbf{y} \in \mathbb{Z}_q^m, a \in \mathbb{Z}_q)\text{:}} \\[4pt]
\quad \text{return } \mathsf{sk_y} := \widetilde{\mathbf{W}}^\top \mathbf{y} - \dfrac{a}{\langle \mathbf{c}, \mathbf{a}^\perp \rangle} \mathbf{a}^\perp \in \mathbb{Z}_q^{k+1} \\[10pt]
\underline{\widetilde{\mathsf{Enc}}(\widetilde{\mathsf{msk}})\text{:}} \\[4pt]
\quad \text{return } ([\mathbf{c}], [\widetilde{\mathbf{W}}\mathbf{c}])
\end{array}
$$

Fig. 16: Simulator $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}})$ from [28] for the one-SEL-SIM security of the single-input scheme for inner product $\mathcal{F}_1^{m,B}$ in Fig. 15.

Proof (of Theorem 5). The simulator $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}})$ are described in 16. Let $\mathcal{A}$ be a PPT adversary, and $\lambda \in \mathbb{N}$ be the security parameter. For all $i \in \{0, \ldots, 2\}$, we define $\mathsf{Adv}_i(\mathcal{A}) := \Pr[\mathsf{Game}_i(1^\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and Game_i.

Game$_0$: is the experiment $\mathbf{REAL}^{\mathcal{FE}}(1^\lambda, \mathcal{A})$.

Game$_1$: is as Game$_0$, except we replace $[\mathbf{Ar}]$ in the challenge ciphertext with $[\mathbf{c}] \leftarrow_\mathrm{R} \mathbb{G}^{k+1} \setminus \mathsf{Span}(\mathbf{A})$, using the $\mathcal{D}_k$-MDDH assumption (cf Lemma 18).

Lemma 18. *There exists a PPT adversary $\mathcal{B}_1$ such that*

$$|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathbb{G}, \mathcal{B}_1}(\lambda) + \frac{1}{q}.$$

Proof. In Game$_1$, we replace the vector $[\mathbf{Ar}]$ in the challenge ciphertext with $[\mathbf{c}] \leftarrow_\mathrm{R} \mathbb{G}^{k+1} \setminus \mathsf{Span}([\mathbf{A}])$. This replacement is justified by the facts that:

1. By the $\mathcal{D}_k$-MDDH assumption, we can switch $([\mathbf{A}], [\mathbf{Ar}])$ to $([\mathbf{A}], [\mathbf{u}])$, where $\mathbf{A} \leftarrow_\mathrm{R} \mathcal{D}_k$, $\mathbf{r} \leftarrow_\mathrm{R} \mathbb{Z}_q^k$, and $\mathbf{u} \leftarrow_\mathrm{R} \mathbb{Z}_q^{k+1}$.
2. The uniform distribution over $\mathbb{G}^{k+1}$ and $\mathbb{G}^{k+1} \setminus \mathsf{Span}(\mathbf{A})$ are $\frac{1}{q}$-close, for any $\mathbf{A} \in \mathbb{Z}_q^{(k+1) \times k}$ of rank k.

Combining these three facts, we obtain a PPT adversary $\mathcal{B}_1$ such that $|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_0(\mathcal{A})| \leq \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathbb{G}, \mathcal{B}_1}(\lambda) + \frac{1}{q}$. $\qquad\square$

Game$_2$: is the experiment $\mathbf{IDEAL}^{\mathcal{FE}}(1^\lambda, \mathcal{A})$. In Lemma 19, we show that Game$_2$ and Game$_1$ are perfectly indistinguishable, using a statistical argument, that crucially relies on the fact that Game$_1$ and Game$_2$ are selective.

Lemma 19. *For all adversaries $\mathcal{B}$, $\mathsf{Adv}_1(\mathcal{B}) = \mathsf{Adv}_2(\mathcal{B})$.*

Proof. In Game$_2$, we perform a change of variables $\widehat{\mathbf{W}} = \mathbf{W} + \frac{1}{\langle \mathbf{c}, \mathbf{a}^\perp \rangle} \mathbf{x}(\mathbf{a}^\perp)^\top$, where $\mathbf{a}^\perp \in \mathbb{Z}_q^{k+1}$ is a non-zero vector such that $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$. This does not change the adversary's view, because $\mathbf{W} \leftarrow_\mathrm{R} \mathbb{Z}_q^{m \times (k+1)}$ is picked *after* the adversary sends its selective challenge $\mathbf{x}$, and therefore, *independently* of it. Since $\mathbf{c} \notin \mathsf{Span}(\mathbf{A})$, we have $\langle \mathbf{c}, \mathbf{a}^\perp \rangle \neq 0$. This way, we have:

$$\mathbf{WA} = \widehat{\mathbf{W}}\mathbf{A}$$
$$\mathbf{x} + \mathbf{Wc} = \widehat{\mathbf{W}}\mathbf{c}$$
$$\mathbf{W}^\top \mathbf{y} = \widehat{\mathbf{W}}^\top \mathbf{y} - \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{c}, \mathbf{a}^\perp \rangle} \mathbf{a}^\perp$$

which coincides precisely with the output of the simulator. This proves $\mathsf{Adv}_2(\mathcal{B}) = \mathsf{Adv}_1(\mathcal{B})$. $\qquad\square$

Putting everything together, we obtain $|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq \mathsf{Adv}_{\mathbb{G},\mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}$. By definition, we have: $\mathsf{Adv}_{\mathcal{A}}^{\mathcal{FE},SEL\text{-}SIM}(\lambda, \mathcal{A}) = |\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})|$ (see Definition 6). Therefore, by the $\mathcal{D}_k$-MDDH assumption in $\mathbb{G}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathcal{FE},SEL\text{-}SIM}(\lambda, \mathcal{A})$ is negligible in λ. $\qquad\square$

Theorem 6 (many-AD-IND security of $\mathcal{FE}$). *If the $\mathcal{D}_k$-MDDH assumption holds in $\mathbb{G}$, then the single-input FE in Figure 15 is many-AD-IND-secure (see Definition 7).*
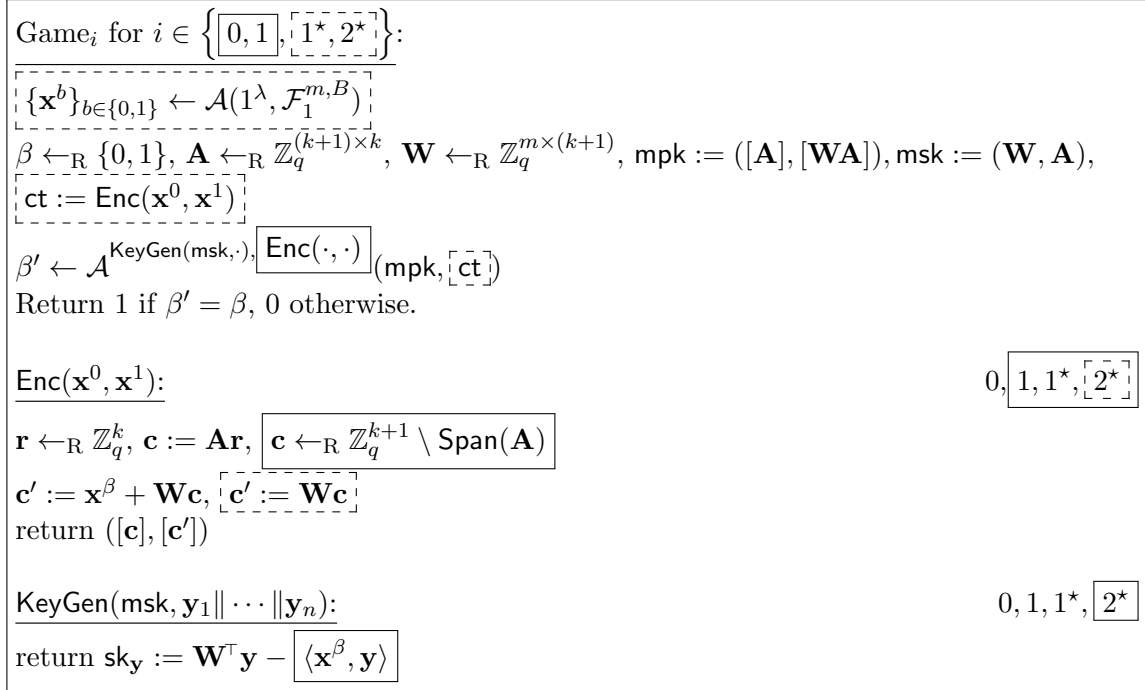
$\boxed{\begin{array}{l}
\underline{\text{Game}_i \text{ for } i \in \left\{\boxed{0,1}, \dashbox{$1^\star, 2^\star$}\right\}:} \\[4pt]
\dashbox{$\{\mathbf{x}^b\}_{b\in\{0,1\}} \leftarrow \mathcal{A}(1^\lambda, \mathcal{F}_1^{m,B})$} \\[4pt]
\beta \leftarrow_R \{0,1\}, \mathbf{A} \leftarrow_R \mathbb{Z}_q^{(k+1)\times k}, \mathbf{W} \leftarrow_R \mathbb{Z}_q^{m\times(k+1)}, \mathsf{mpk} := ([\mathbf{A}], [\mathbf{WA}]), \mathsf{msk} := (\mathbf{W}, \mathbf{A}), \\[4pt]
\dashbox{$\mathsf{ct} := \mathsf{Enc}(\mathbf{x}^0, \mathbf{x}^1)$} \\[4pt]
\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot), \boxed{\mathsf{Enc}(\cdot,\cdot)}}(\mathsf{mpk}, \dashbox{ct}) \\[4pt]
\text{Return 1 if } \beta' = \beta, \text{0 otherwise.} \\[10pt]
\underline{\mathsf{Enc}(\mathbf{x}^0, \mathbf{x}^1):} \hfill 0, \boxed{1, 1^\star, \dashbox{$2^\star$}} \\[4pt]
\mathbf{r} \leftarrow_R \mathbb{Z}_q^k, \mathbf{c} := \mathbf{Ar}, \boxed{\mathbf{c} \leftarrow_R \mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A})} \\[4pt]
\mathbf{c}' := \mathbf{x}^\beta + \mathbf{Wc}, \dashbox{$\mathbf{c}' := \mathbf{Wc}$} \\[4pt]
\text{return } ([\mathbf{c}], [\mathbf{c}']) \\[10pt]
\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}_1\|\cdots\|\mathbf{y}_n):} \hfill 0, 1, 1^\star, \boxed{2^\star} \\[4pt]
\text{return } \mathsf{sk}_{\mathbf{y}} := \mathbf{W}^\top \mathbf{y} - \boxed{\langle \mathbf{x}^\beta, \mathbf{y} \rangle}
\end{array}}$

Fig. 17: Game_i for $i \in \{0, 1, 2\}$, for the proof of Theorem 6. In each procedure, the components inside a solid (dotted) frame are only present in the games marked by a solid (dotted) frame.

Proof (of Theorem 6). Since the FE scheme $\mathcal{FE}$ is public-key, it suffices to show one-AD-IND security. We proceed via a series of Game_i for $i \in \{0, \ldots, 3\}$, described in Fig. 17. Let $\mathcal{A}$ be a PPT adversary, and $\lambda \in \mathbb{N}$ be the security parameter. For all $i \in \{0, \ldots, 3\}$, we define $\mathsf{Adv}_i(\mathcal{A}) := \Pr[\text{Game}_i(1^\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and Game_i.

Game$_0$: is the experiment **one-AD-IND**$^{\mathcal{FE}}(1^\lambda, \mathcal{A})$ (see Definition 7). In particular, $\mathsf{Adv}_0(\mathcal{A}) = \mathsf{Adv}^{\mathcal{MIFE},one\text{-}AD\text{-}IND}(\lambda, \mathcal{A})$.

Game$_1$: we change the distribution of the vector $[\mathbf{c}]_1$ computed by $\mathsf{Enc}(\mathbf{x}^0, \mathbf{x}^1)$, using the $\mathcal{D}_k$-MDDH assumption (cf Lemma 20).

Lemma 20 (Game 0 to 1). *There exists a PPT adversary $\mathcal{B}_1$ such that*

$$\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \le \mathsf{Adv}_{\mathbb{G},\mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}.$$

Proof. In Game_1, we replace the vector $[\mathbf{Ar}]$ in the challenge ciphertext with $[\mathbf{c}] \leftarrow_{\mathrm{R}} \mathbb{G}^{k+1}\backslash\mathsf{Span}(\mathbf{A})$. This replacement is justified by the facts that:

1. By the $\mathcal{D}_k$-MDDH assumption, we can switch $([\mathbf{A}], [\mathbf{Ar}])$ to $([\mathbf{A}], [\mathbf{u}])$ for $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k$, $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k+1}$.
2. The uniform distribution over $\mathbb{G}^{k+1}$ and $\mathbb{G}^{k+1} \setminus \mathsf{Span}(\mathbf{A})$ are $\frac{1}{q}$-close, for $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k$ of rank k.

Combining these facts, we obtain a PPT adversary $\mathcal{B}$ such that $\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \le \mathsf{Adv}_{\mathbb{G}_s,\mathcal{B}}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}$. $\qquad\square$

Next, we prove that $\mathsf{Adv}_1(\mathcal{A}) = 0$. To do so, we first consider the selective variant of Game_1, which we call $\mathrm{Game}_{1\star}$ (it is defined in Fig. 17, and the one-SEL-IND security is defined in Definition 5). We prove that no adversary can win $\mathrm{Game}_{1\star}$, using a hybrid $\mathrm{Game}_{2\star}$. That is, we prove that $\mathrm{Game}_{1\star}$ and $\mathrm{Game}_{2\star}$ are perfectly indistinguishable in Lemma 22, and we prove that no adversary can win $\mathrm{Game}_{2\star}$ in Lemma 23. Finally, we use complexity leveraging to obtain perfect security of the adaptive Game_1, in Lemma 21.

$\mathrm{Game}_{1\star}$: is the selective variant of Game_1, that is, any adversary playing this game has to commit to its challenge queries $(\mathbf{x}^0, \mathbf{x}^1)$ beforehand. In the following lemma, we build an adversary $\mathcal{B}_{1\star}$ against $\mathrm{Game}_{1\star}$, such that $\mathsf{Adv}_1(\mathcal{A}) \le q^{2m} \cdot \mathsf{Adv}_{1\star}(\mathcal{B}_{1\star})$. Note that this exponential security loss does not ruin security, since we will prove that $\mathsf{Adv}_{1\star}(\mathcal{B}_{1\star}) = 0$.

Lemma 21 (Game$_1$ to Game$_{1\star}$). *We build a PPT adversary $\mathcal{B}_{1\star}$ such that $\mathsf{Adv}_1(\mathcal{A}) \le q^{2m} \cdot \mathsf{Adv}_{1\star}(\mathcal{B}_{1\star})$.*

Proof. $\mathrm{Game}_{1\star}$ is exactly the same than Game_1 except that the former is selective: any adversary playing $\mathrm{Game}_{1\star}$ has to commits to its challenge $(\mathbf{x}^0, \mathbf{x}^1)$ beforehand. We use complexity leveraging to build an adversary in the selective $\mathrm{Game}_{1\star}$ from the adversary $\mathcal{A}$ in the adaptive Game_1 (these game are described in Fig. 17).

The adversary $\mathcal{B}_{1\star}$ against $\mathrm{Game}_{1\star}$ first guesses the future queries $(\mathbf{x}^0, \mathbf{x}^1)$ of $\mathcal{A}$, aborts (and loses the game) if the guess is unsuccessful; otherwise, it behaves exactly as $\mathcal{A}$ (it does the same queries to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$). It is clear that the advantage of $\mathcal{B}_{1\star}$ in $\mathrm{Game}_{1\star}$ is at least $q^{-2m} \cdot \mathsf{Adv}_1(\mathcal{A})$. $\quad\square$

$\mathrm{Game}_{2\star}$: is like $\mathrm{Game}_{1\star}$, except that $\mathsf{Enc}(\mathbf{x}^0, \mathbf{x}^0)$ computes $[\mathbf{c}']_1 := \mathbf{W}[\mathbf{c}]_1$, and $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$ computes $\mathsf{sk}_{\mathbf{y}} := \mathbf{W}^\top \mathbf{y} - \langle \mathbf{x}^\beta, \mathbf{y} \rangle$. We show in Lemma 22 that $\mathrm{Game}_{1\star}$ and $\mathrm{Game}_{2\star}$ are perfectly indistinguishable, using an information theoretic argument that crucially relies on the fact that $\mathrm{Game}_{1\star}$ and $\mathrm{Game}_{2\star}$ are *selective*. Then, in Lemma 23, we show that $\mathrm{Game}_{2\star}$ is independent of β, that is, no adversary can win with non zero advantage in this game. This relies on the fact that in $\mathrm{Game}_{2\star}$, β only shows up in the value $\langle \mathbf{x}^\beta, \mathbf{y} \rangle$, which is equal $\langle \mathbf{x}^0, \mathbf{y} \rangle$, as required by the ideal functionality.

Lemma 22 (Game$_{1\star}$ to Game$_{2\star}$). *For all adversaries* $\mathcal{B}$, $\mathsf{Adv}_{1\star}(\mathcal{B}) = \mathsf{Adv}_{2\star}(\mathcal{B})$.

Proof. We perform the change of variables $\mathbf{W}' := \mathbf{W} - \frac{1}{\langle \mathbf{c}, \mathbf{a}^\perp \rangle} \mathbf{x}^\beta (\mathbf{a}^\perp)^\top$, where $\mathbf{W}' \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m \times (k+1)}$, $\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k+1} \setminus \mathsf{Span}(\mathbf{A})$, and where $\mathbf{a}^\perp \in \mathbb{Z}_q^{k+1}$ is a non-zero vector such that $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$. Note that since $\mathbf{c} \notin \mathsf{Span}(\mathbf{A})$, we have $\langle \mathbf{c}, \mathbf{a}^\perp \rangle \neq 0$. This does not change the adversary's view, because $\mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{m \times (k+1)}$ is picked *after* the adversary sends its selective challenge $(\mathbf{x}^0, \mathbf{x}^1)$, and therefore, *independently* of it (note that here we crucially rely on the fact the Game$_{1\star}$ and Game$_{2\star}$ are *selective*).

This way, we have:

$$\mathbf{W}'\mathbf{A} = \mathbf{W}\mathbf{A}$$
$$\mathbf{x}^\beta + \mathbf{W}'\mathbf{c} = \mathbf{W}\mathbf{c}$$
$$\mathbf{W}'^\top \mathbf{y} = \mathbf{W}^\top \mathbf{y} - \langle \mathbf{x}^\beta, \mathbf{y} \rangle \cdot \mathbf{a}^\perp$$

as in Game$_{2\star}$. $\qquad\square$

Lemma 23 (Game$_{2\star}$). *For all adversaries* $\mathcal{B}$, $\mathsf{Adv}_{2\star}(\mathcal{B}) = 0$.

Proof. We use the fact that for the query $(\mathbf{x}^0, \mathbf{x}^1)$ to Enc, and for all queries $\mathbf{y}$ to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, we have, by definition of the security game:

$$\langle \mathbf{x}^0, \mathbf{y} \rangle = \langle \mathbf{x}^1, \mathbf{y} \rangle.$$

Therefore, for each call $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y})$, the secret key $\mathsf{sk}_{\mathbf{y}}$, which is of the form $\mathsf{sk}_{\mathbf{y}} := \mathbf{W}^\top \mathbf{y} - \langle \mathbf{x}^\beta, \mathbf{y} \rangle$, is independent of β. Since the challenge ciphertext and the public key are also independent of β, we haves $\mathsf{Adv}_{2\star}(\mathcal{B}) = 0$. $\qquad\square$

Summing up, we have shown in the previous lemmas that for all PPT adversaries $\mathcal{A}$ and $\lambda \in \mathbb{N}$ we have the following.

- In Lemma 20, we build a PPT adversary $\mathcal{B}_1$ such that $\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}, \mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}$.
- In Lemma 21, we build a PPT adversary $\mathcal{B}_{1\star}$ such that $\mathsf{Adv}_1(\mathcal{A}) \leq q^{-2m}\mathsf{Adv}_{1\star}(\mathcal{B}_{1\star})$, using a complexity leveraging argument.
- In Lemma 22, we show that for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{1\star}(\mathcal{B}) = \mathsf{Adv}_{2\star}(\mathcal{B})$, using a statistical argument.
- In Lemma 23, we show that for all adversaries $\mathcal{B}$, $\mathsf{Adv}_{2\star}(\mathcal{B}) = 0$, using the restriction on queries to KeyGen and Enc.

Combining everything together, we obtain:

$$\mathsf{Adv}_0(\mathcal{A}) \leq \mathsf{Adv}_{\mathbb{G}, \mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + \frac{1}{q}.$$

Using the $\mathcal{D}_k$-MDDH assumption in $\mathbb{G}$, this means $\mathsf{Adv}_0(\mathcal{A})$ is a negligible function of λ. $\qquad\square$

B   Motivating applications

Suppose we have a company which keeps profiles of its employees including the grades for their skills that they received in their last evaluation. This information is considered private and the only person who has access to such information about the employees is their direct manager. When a new project is started at the company, a lead manager is assigned to the project and she has to form a new team. In order to evaluate a possible team configuration, she needs to evaluate the skills of the team as a whole. This can be achieved by obtaining a score for the team that is the weighted sum of the skills of people serving in different positions in the team. The specific weights assigned to various skills are determined by the needs of the particular project.

A MIFE scheme can provide a solution for the above scenario as follows. The grades for the skills of each employee can be represented as an integer vector, which will be encrypted under a MIFE scheme that uses its encryption slots to represent different team positions. When a new project is established, the lead manager is granted a decryption key that assigns weights to each of the skills of different team members. She can use this key to evaluate various combinations of people for the team while learning nothing more about everyone's profile than the total team score.

A similar example is the construction of a complex machine that requires parts from different manufacturers. Each part is rated based on different quality characteristics and price, which are all manufacturer's proprietary information until a contract has been signed. The ultimate goal is to assemble a construction of parts that achieve a reasonable trade-off between quality and price. In order to evaluate different construction configurations, the company wants to compute cumulative score for each configuration that is a weighted sum over the quality rates and price of each of the parts.

C   Additional Definitions

Here we provide many-AD-IND security definition for MIFE in the public key setting where the adversary holds encryptions keys for a subset of the MIFE encryption slots. We first define the MIFE functionality that allows public encryption keys for some slots

Definition 8 (Multi-input Function Encryption). *Let $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be an ensemble where each $\mathcal{F}_n$ is a family of n-ary functions. A function $f \in \mathcal{F}_n$ is defined as follows $f : \mathcal{X}_1 \times \ldots \times \mathcal{X}_n \to \mathcal{Y}$. A multi-input functional encryption scheme $\mathcal{MIFE}$ for $\mathcal{F}$ consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda, \mathcal{F}_n)$: *on input the security parameter λ and a description of $\mathcal{F}_n \in \mathcal{F}$ outputs n encryption keys $\mathsf{ek}_1, \ldots, \mathsf{ek}_n$ and a master secret key msk.*
- $\mathsf{Enc}(\mathsf{ek}_i, i, m)$: *on input $\mathsf{ek}_i \in \{\mathsf{ek}_1, \ldots, \mathsf{ek}_n\}$ and a message $m \in \mathcal{X}_i$ outputs a ciphertext ct for the i-th MIFE slot. We assume that each ciphertext has an associated index i, which denotes what slot this ciphertext can be used for.*
- $\mathsf{KeyGen}(\mathsf{msk}, f)$: *on input the master secret key msk and a function $f \in \mathcal{F}_n$ outputs a decryption key sk_f that takes inputs encrypted under $\mathsf{ek}_1, \ldots \mathsf{ek}_n$.*
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$: *on input a decryption key for function f and sk_f ciphertexts outputs a string $y \in \mathcal{Y}$.*

The scheme $\mathcal{MIFE}$ *is correct if for all* $f \in \mathcal{F}$ *and all* $x_i \in \mathcal{X}_i$ *for* $1 \le i \le n$, *we have*

$$\Pr\left[\begin{array}{r}(\mathsf{ek}_1,\ldots,\mathsf{ek}_n,\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda,n); \\ \mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk},f); \\ \mathsf{Dec}(\mathsf{sk}_f,\mathsf{Enc}(\mathsf{ek}_1,x_1),\ldots,\mathsf{Enc}(\mathsf{ek}_n,x_n)) = f(x_1,\ldots,x_n)\end{array}\right] = 1,$$

where the probability is taken over the coins of Setup, KeyGen *and* Enc.

For out definition we will use the notion of *I-compatibility*, which is defined next.

Definition 9 (*I*-Compatibility). *Let* $\mathcal{F}_n = \{f\}$ *be a family of n-ary functions and let* $N = \{1,\ldots,n\}$, $I \subset N$. *The pairs of input vectors* $\mathbf{x}_0 = \left(\{x_1^{j,0}\}_{j=1}^{q_1},\ldots,\{x_n^{j,0}\}_{j=1}^{q_n}\right)$ *and* $\mathbf{x}_1 = \left(\{x_1^{j,1}\}_{j=1}^{q_1},\ldots,\{x_n^{j,1}\}_{j=1}^{q_n}\right)$ *are I-compatible if they satisfy the following property:*

For every $f \in \{f\}$, *every* $I' = \{i_1,\ldots,i_t\} \subseteq I \cup \emptyset$, *every* $j_1 \in [q_1],\ldots,j_{n-t} \in [q_n]$, *for every* $x'_{i_1} \in \mathcal{X}_{i_1},\ldots,x'_{i_t} \in \mathcal{X}_{i_t}$ *the following two distributions are indistinguishable:*

$$f\left(\pi(x_{i_1}^{j_1,0},\ldots,x_{i_{n-t}}^{j_{n-t},0},x'_{i_1},\ldots,x'_{i_t})\right) = f\left(\pi(x_{i_1}^{j_1,1},\ldots,x_{i_{n-t}}^{j_{n-t},1},x'_{i_1},\ldots,x'_{i_t})\right),$$

where $\pi(y_{i_1},\ldots,y_{i_n})$ *denotes a permutation of the values* $y_{i_1},\ldots,y_{i_n}$ *such that the value* y_{i_j} *is mapped to the l'th location of* y_{i_j} *is the l'th input (out of n input) to* f.

Using the above notions of *I*-compatibility the indistinguishability security for MIFE is defined as follows.

Definition 10 (many-AD-IND-secure MIFE). *A multi-input functional encryption* $\mathcal{MIFE}$ *for n-ary functions* $\mathcal{F}$ *is* (t,q)-IND *secure if for every PPT adversary* $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2,\mathcal{A}_3)$, *the advantage of* $\mathcal{A}$ *defined as*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathcal{MIFE},IND} = |\Pr[\mathbf{IND}_{\mathcal{A},1}^{\mathcal{MIFE}}(1^\lambda) = 1] - \Pr[\mathbf{IND}_{\mathcal{A},0}^{\mathcal{MIFE}}(1^\lambda) = 1]|$$

is negligible and the experiment $\mathbf{IND}_{\mathcal{A},\beta}^{\mathcal{MIFE}}(1^\lambda)$ *is defined as follows:*

$\underline{\textit{Experiment}\ \mathbf{IND}_{\mathcal{A},\beta}^{\mathcal{MIFE}}(1^\lambda):}$

$(I,\mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda)$ *where* $|I| = t$

$(\{\mathsf{ek}_i\}_{i\in[n]},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$(\mathbf{X}^0,\mathbf{X}^1,\mathsf{st}_1) \leftarrow \mathcal{A}_1^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{st}_0,\{\mathsf{ek}_i\}_{i\in I})$ *for* $\left\{\mathbf{X}^d = \{x_1^{j,d},\ldots,x_n^{j,d}\}_{j=1}^{q}\right\}_{d=1,2}$

$\mathsf{ct}_{i,j} \leftarrow \mathsf{Enc}(\mathsf{ek}_i,x_i^{j,\beta})\ \forall i \in [n], j \in [q]$

$\beta' \leftarrow \mathcal{A}_2^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{st}_1,\mathbf{ct})$

Output: β'

In the above experiment we require that all queries for $\{f\}$ *for decryption functions made by* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *must be I-compatible with* $\mathbf{X}^0$ *and* $\mathbf{X}^1$ *according to Definition 9.*

We call the MIFE scheme public key many-AD-IND-secure when $t = n$, *i.e., the adversary obtain the encryption keys for all slots.*

Since our paper focuses on multi-input functional encryption for inner product functions, we note that the I-compatibility condition in this setting becomes as follows. Each function f_i is defined by a vector $f = (\mathbf{y}_1 \| \cdots \| \mathbf{y}_n)$. Then, if $\overline{I}$ is the set of the inputs for which the adversary does not receive encryption keys, we require that for every $j_1, j_1' \in [q_1], \ldots, j_{n-t}, j_{n-t}' \in [q_n]$ and every $J \subseteq [n]$ such that $\overline{I} \subseteq J$:

$$\sum_{i \in J} \langle \mathbf{x}_i^{j_i, 0}, \mathbf{y}_i \rangle = \sum_{i \in J} \langle \mathbf{x}_i^{j_i', 1}, \mathbf{y}_i \rangle.$$

This restriction is due to the fact that the adversary can always encrypt the all 0's vectors in the slots for which it holds encryption keys. Thus, if the above equality does not hold, the adversary will be able to distinguish the challenges trivially.

We note that in the case of public key MIFE, the above restriction amounts to

$$\langle \mathbf{x}_i^{j_i, 0}, \mathbf{y}_i \rangle = \langle \mathbf{x}_i^{j_i', 1}, \mathbf{y}_i \rangle \quad \forall i \in [n],$$

since $\overline{I} = \emptyset$ and the sets J include every singleton set.

D   Extensions

D.1   Public-Key MIFE for Inner Product

This is easy, as discussed in the introduction. Run n independent copies of the single-input scheme; use the i'th copy to encrypt $\mathbf{x}_i$ in the i'th slot; and the new secret key is the collection of the n secret keys corresponding to each of $\mathbf{y}_1, \ldots, \mathbf{y}_n$. Decryption recovers $\langle \mathbf{x}_1, \mathbf{y}_1 \rangle, \ldots \langle \mathbf{x}_n, \mathbf{y}_n \rangle$ and returns the sum of these values. This means that the adversary also learns each of $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$ but that is inherent leakage from the ideal functionality. Concretely, an adversary can always pad an encryption of $\mathbf{x}_i$ in the i'th slot with encryptions of $\mathbf{0}$'s in the remaining $n-1$ slots and then decrypt with a key for $\mathbf{y}_1, \ldots, \mathbf{y}_n$ to learn $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$. Security is immediate since the underlying FE guarantees that there is no leakage beyond $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$.

D.2   Impossibility for many-SEL-SIM.

The impossibility result for simulation-based IBE in [11, 3] applies to this setting, even for private-key schemes with "selective" security in the single-input setting for one-dimensional inner product over bits. Suppose the secret key has ℓ bits. The adversary starts by requesting for $\ell + \lambda$ encryptions of random bits. At this point, the simulator needs to simulate these $\ell + \lambda$ ciphertexts without having learned anything from the ideal functionality. The adversary then requests for a secret key for 1, and the simulator needs to produce an ℓ-bit secret key that decrypts the simulated ciphertext to an arbitrary sequence of $\ell + \lambda$ bits. This is impossible even for a computationally unbounded adversary.