Tightly Secure Inner Product Functional Encryption: Multi-Input and Function-Hiding Constructions

Junichi Tomida[*]
NTT Corporation, Japan

Abstract

Tightly secure cryptographic schemes have been extensively studied in the fields of chosen-ciphertext secure public-key encryption (CCA-secure PKE), identity-based encryption (IBE), signatures and more. We extend tightly secure cryptography to inner product functional encryption (IPFE) and present the first tightly secure schemes related to IPFE.

We first construct a new IPFE scheme that is tightly secure in the multi-user and multi-challenge setting. In other words, the security of our scheme does not degrade even if an adversary obtains many ciphertexts generated by many users. Our scheme is constructible on a pairing-free group and secure under the matrix decisional Diffie-Hellman (MDDH) assumption, which is the generalization of the decisional Diffie-Hellman (DDH) assumption. Applying the known conversions by Lin (CRYPTO 2017) and Abdalla et al. (CRYPTO 2018) to our scheme, we can obtain the first tightly secure function-hiding IPFE scheme and multi-input IPFE (MIPFE) scheme respectively.

Our second main contribution is the proposal of a new generic conversion from function-hiding IPFE to function-hiding MIPFE, which was left as an open problem by Abdalla et al. (CRYPTO 2018). We can obtain the first tightly secure function-hiding MIPFE scheme by applying our conversion to the tightly secure function-hiding IPFE scheme described above.

Finally, the security reductions of all our schemes are fully tight, which means that the security of our schemes is reduced to the MDDH assumption with a constant security loss.

Keywords: functional encryption, inner product, tight security, multi-input, function-hiding

1 Introduction

(Multi-input) inner product functional encryption. Functional encryption (FE) [13, 37] is a relatively novel cryptographic notion that has a crucially different feature from traditional encryption schemes. Specifically, FE schemes allow us to obtain computation results from encrypted data without revealing any other information about the underlying data. This is in contrast to traditional encryption schemes, in which only owners of legitimate keys can learn entire underlying data from ciphertexts while others can learn nothing. An FE scheme supports a certain function class $\mathcal{F}$ and in which an owner of a master secret can issue a secret key sk_f for any function $f \in \mathcal{F}$. Decryption of a ciphertext ct_x of message x with sk_f yields the computation result $f(x)$ and nothing else.

Multi-input functional encryption (MIFE) [28] is a natural extension of FE, which can handle a function class that takes multiple inputs. Roughly speaking, an owner of sk_f can learn the computation result $f(x_1, \ldots, x_\mu)$ from ciphertexts $\mathsf{ct}_{x_1}, \ldots, \mathsf{ct}_{x_\mu}$ of messages $x_1, \ldots, x_\mu$ for some natural number $\mu > 2$.

Known (MI)FE schemes can be classified into two categories with respect to their function classes.

[*]junichi.tomida.vw@hco.ntt.co.jp

General functionalities: This category consists of (MI)FE schemes for general circuits, e.g., [8, 23, 24, 28, 39]. Although they are powerful enough to handle all functions computable in polynomial time, known schemes are built on quite heavy cryptographic primitives such as indistinguishability obfuscation [23] or multi-linear maps [22]. Thus, they are captured as rather feasibility results.

Specific functionalities: The second category covers (MI)FE schemes for specific functions such as inner product and quadratic function, e.g., [2, 4, 6, 9]. They are aimed at obtaining more practical features, namely, efficiency and concrete security, with sacrificing the generality. Therefore, most of them have simple constructions, and their security is based on standard assumptions.

Inner product functional encryption (IPFE) [2] and multi-input IPFE (MIPFE) [4], categorized into the latter, are FE and MIFE respectively, whose function classes are inner product. More precisely, in an (M)IPFE scheme, a secret key $\mathsf{sk}_{\mathbf{y}_1,\dots,\mathbf{y}_\mu}$ is associated with vectors $\mathbf{y}_1, \dots, \mathbf{y}_\mu$, and decrypting ciphertexts $\mathsf{ct}_{\mathbf{x}_1}, \dots, \mathsf{ct}_{\mathbf{x}_\mu}$ of vectors $\mathbf{x}_1, \dots, \mathbf{x}_\mu$ with $\mathsf{sk}_{\mathbf{y}_1,\dots,\mathbf{y}_\mu}$ reveals the summation of the inner products $\sum_{i\in[\mu]}\langle \mathbf{x}_i, \mathbf{y}_i\rangle$. When $\mu = 1$, the above description corresponds to an IPFE scheme. Inner product is a simple but powerful functionality, and many practical applications of IPFE have been suggested, e.g, biometric authentication, nearest-neighbor search and statistical analysis [2, 32].

Function privacy. In (MI)FE, we can consider two types of privacy: message privacy and function privacy. Message privacy, which is essential for standard (MI)FE schemes, is the property that ciphertexts do not reveal any information about underlying data. On the other hand, function privacy is an additional but important property for (MI)FE schemes, which indicates that secret keys also hide the information of the corresponding function. Function privacy is essential for some applications such as delegation of sensitive computation [15]. We often call (MI)FE with function privacy as function-hiding (MI)FE. Function-hiding (MI)FE schemes have also been studied for both general functionalities [14, 15] and specific functionalities [12, 18, 32, 38].

Tight security. When we try to prove the security of a cryptographic scheme, we often construct a reduction algorithm that solves a problem assumed to be hard by utilizing a PPT adversary that breaks the security of the scheme. Then, breaking the security of the scheme immediately implies solving the hard problem. It is both theoretically and practically important to evaluate how difficult breaking the scheme is compared with solving the problem. More formally, when the reduction algorithm equipped with an adversary that breaks the scheme with probability ϵ in time t solves the underlying problem with probability ϵ/L in roughly the same time t, it is important to evaluate the security loss L. This is because we need to set the parameter size of the scheme large enough to negate the effect of L for the security guarantee. Thus, the smaller the security loss L, the more desirable the security reduction. We say that the security reduction is tight if the security loss is constant, i.e., $L = O(1)$.

When we consider public-key primitives such as public-key encryption (PKE) or identity-based encryption (IBE), we usually prove their security in the single-challenge setting. This is because the security of public-key primitives in the single-challenge setting normally implies that in the multi-user and multi-challenge setting via hybrid argument, which is more realistic setting where an adversary can make polynomially many challenge queries against multiple users. However, such a hybrid argument increases the security loss by the factor of μq, where μ is the number of users and q is the maximum number of challenge queries for each users [11]. Since it is difficult to assume the numbers of users and ciphertexts that will be involved with the scheme at deployment time, we strongly desire cryptographic schemes whose security is guaranteed independently of those numbers.

Motivated by the above reason, (almost) tightly secure cryptographic schemes have been extensively studied in various fields, especially on chosen-ciphertext secure PKE (CCA-secure PKE), IBE, and signature, e.g, [7, 17, 25, 26, 29–31, 33]. In spite of such a great deal of effort, tightly secure schemes in the context of advanced encryption are known only for IBE except the very recent result on broadcast encryption by Gay et al. [27]. Hence, it is an important and interesting task to explore what kind of cryptographic schemes can achieve tight security.

Tight security for IPFE. We would like to discuss the importance of tightly secure IPFE in more detail. We consider that the most significant situation where we need a tightly secure IPFE scheme is when a function-hiding scheme is needed. This is because the only way that we know to realize function-hiding IPFE schemes requires bilinear groups, which is relatively susceptible to security loss. The one solution to compensate for security loss caused by loose reduction is to increase the parameter size of underlying primitives, e.g., bilinear groups, which will reinforce the difficulty of underlying problems, e.g., the matrix Diffie-Hellman problem. As observed by Abe et al. [5], however, this is not an easy task for bilinear groups because there are many factors that involve the security and efficiency of them such as the choice of curves, pairings, and various parameters like embedding degrees. Hence, we typically adopt one from existing well-studied settings, which are investigated only for standard parameters such as 128, 192, and 256-bit security. The main problem of this fact is that there is no intermediate instantiation among these parameters, and one have to hop to the next standard level if stronger security is necessary. A pairing computation is especially influenced by this hop; for instance, they state that a pairing in the 192-bit security takes 6 to 7 times more time than in the 128-bit security on ordinary personal computers [10, 20].

Additionally, it is not unrealistic that an adversary obtains a large amount of ciphertexts so that we need to consider the security loss of IPFE schemes. Let us consider the case to use a function-hiding IPFE scheme for DNA analysis. Suppose a national institution holds a database consisting of a certain part of the human's DNA sequence. It is rational to assume that the part consists of 2^{13} bases and the number of the samples is 2^{20}; actually, GenBank operated by the National Center for Biotechnology Information has more than 2^{27} sequences [1]. Each sample is encoded to a binary vector setting as A=(1,0,0,0), T=(0,1,0,0), and so on, and stored in a cloud server with an encrypted form. We can check the number of the same bases between encrypted sequences and a target sequence by decrypting with a secret key for the target sequence. Because DNA sequences have a correlation with phenotypes, the DNA similarity check will be useful for genetical research, medical diagnosis, etc. We need the function-hiding property because target sequences are also personal data and thus sensitive. In this situation, the possibly untrusted server has $q = 2^{20}$ ciphertexts, large enough to consider the security loss of the scheme. Decryption of all known schemes involves the same number of pairings as the order of the vector length: $m = 2^{15}$ per one sample in our case. Thus, the choice of the security level significantly affects the efficiency of the system, and we can conclude that tight security is a very important concept in the context of IPFE as well as other cryptosystems.

1.1 Our Contributions

We extend the realm of tightly secure cryptography to IPFE and present a series of the first tightly secure (M)IPFE schemes. Our first main contribution is to construct the first tightly secure public-key IPFE scheme in the multi-user and multi-challenge setting. Note that previous IPFE schemes are tightly reduced to underlying assumptions in the single-challenge setting [6], which means that their security is independent from the number of secret key queries. To our knowledge, however, there are no results on tight security of IPFE in the multi-user and multi-challenge setting. Our tightly secure IPFE scheme is constructible from a pairing-free group and its security is based on the matrix decisional Diffie-Hellman (MDDH) assumption, which is a generalization of the well-studied decisional Diffie-Hellman (DDH) assumption, with a very small constant security loss.

Our result can be easily extended to the multi-input setting. Recently, Abdalla et al. proposed a generic conversion from an IPFE scheme into a MIPFE scheme [3, 4]. Their conversion employs parallel execution of μ instances of the underlying IPFE scheme that is secure in the multi-challenge setting. By this construction, their conversion incurs a security loss of $O(\mu q)$ if we apply it to an IPFE scheme that is secure in the single challenge setting, where μ is the number of slots of the converted scheme and q is the maximum number of adversary's ciphertext queries for each slot. Interestingly, this construction is precisely compatible with an IPFE scheme that is secure in the multi-user and multi-challenge setting. In other words, the security of the converted MIPFE scheme is tightly reduced to that of the underlying IPFE scheme if the underlying scheme is secure in the multi-user and multi-challenge setting. Additionally, our scheme satisfies the requirement for the conversion. Thus, we can obtain the first tightly secure MIPFE scheme.

Another important issue is the realization of tightly secure function-hiding (M)IPFE schemes. All previous function-hiding schemes suffer from a security loss of $L = O(q_{\mathsf{ct}} + q_{\mathsf{sk}})$, where q_{ct} (resp. q_{sk}) refers to the total number of ciphertext (resp. secret key) queries [12, 18, 34, 38]. To achieve tight security, we utilize Lin's technique, who presented a simple paradigm to construct a function-hiding (private-key) IPFE scheme from a (public-key) IPFE scheme [34]. Applying her paradigm to our IPFE scheme, we can obtain the first tightly secure function-hiding IPFE scheme that is based on bilinear groups. However, the naive application of her paradigm to our scheme results in a redundant scheme. Thus, we optimize the scheme by reducing the unnecessary part.

The final target is to construct a tightly secure function-hiding MIPFE scheme. Unfortunately, there is no known generic technique to achieve a function-hiding MIPFE scheme. In fact, Abdalla et al. mention that a powerful conversion to achieve a function-hiding MIPFE scheme is a very interesting open problem [3]. Furthermore, the techniques used in the rather specific constructions of known function-hiding MIPFE schemes [3, 19] are not applicable to our situation. Roughly speaking, this is because our scheme requires the selective setting in a certain step of the proof, if we naively try to prove the security similarly to [3, 19].

Our second main contribution is overcoming this problem by solving the open problem posed by Abdalla et al., that is, we introduce a new powerful and generic conversion. It converts a (weakly) function-hiding IPFE scheme into a (fully) function-hiding MIPFE scheme. Our conversion is as general as that for constructing non-function-hiding MIPFE by Abdalla et al. [3]: the requirements for an underlying scheme are essentially the same. Hence, if new function-hiding IPFE schemes are proposed in the future, e.g., based on lattices, we may utilize our conversion to obtain new function-hiding MIPFE schemes though some modification will be necessary. Additionally, we can obtain (non-tightly-secure) function-hiding MIPFE schemes in a more modular way than the previous ones [3, 19] by utilizing our conversion to function-hiding IPFE schemes, e.g., the scheme from AGRW17 [4] + Lin17 [34] (Appendix A). Applying our conversion to our tightly secure function-hiding IPFE scheme, we can finally achieve the first tightly secure function-hiding MIPFE scheme.

Similarly to all previous IPFE schemes based on a cyclic group or bilinear groups, the decryption algorithms of our schemes require to solve the discrete logarithm problem on a decryption value. As pointed out in [2, 32], however, this step is not so problematic in many cases. This is mainly because decryption values will not become exponentially large in real applications. Additionally, although there are some IPFE schemes that allow exponentially large outputs, they are either inefficient due to the large modulus [6] or based on a non-standard assumption [16].

We summarize the comparison of our schemes with previous ones in Tables 1 to 4. In these tables, we count the numbers of elements assuming that a matrix distribution $\mathcal{D}_k$ is a uniform one over $\mathbb{Z}_p^{(k+1)\times k}$. Some readers may be concerned about the increase of the key and ciphertext sizes, which may slow the efficiency of the system even after the compensation of security loss. However, we would like to emphasize that our contribution is a theoretically and technically significant step in tightly secure cryptography. Furthermore, our schemes may outperform previous ones in some situations. For example, when we instantiate our function-hiding IPFE scheme from the SXDH, it takes almost 5 times more pairings in decryption than the state-of-the-art scheme (Table 3). As discussed in the previous subsection, the difference of security level possibly affects pairings by the factor of 6 to 7 in practice, and thus there is a possibility that the decryption, the most important process of IPFE, of our scheme is faster than those of previous ones in the same security level. We leave constructing more compact tightly secure IPFE schemes as an interesting open problem.

2 Technical Overview

In this section, we briefly explain our novel techniques. We write this section assuming that readers are familiar with the notations and notions explained in Section 3. Refer to Section 3 if any notations and notions are unfamiliar.

2.1 Tightly Secure IPFE

Our scheme is secure in the multi-user and multi-challenge setting under the MDDH assumption, but here we describe our scheme based on the DDH assumption in the single-user and multi-challenge setting to ease the exposition. Our starting point is the adaptively secure IPFE scheme by Agrawal et al. [6]. We briefly describe their scheme below. Let m be a vector length in the scheme.

$\mathsf{Setup}(1^\lambda, 1^m)$**:** $a \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p,\ \ \mathbf{W} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{m\times 2},\ \ \mathbf{a} := (a, 1),\ \ \mathsf{pk} := ([\mathbf{a}], [\mathbf{Wa}]),\ \ \mathsf{msk} := \mathbf{W}.$

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** $s \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p,\ \ \mathsf{ct} := ([s\mathbf{a}], [s\mathbf{Wa} + \mathbf{x}]).$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y})$**:** $\mathsf{sk} := (-\mathbf{W}^\top \mathbf{y}, \mathbf{y}).$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk})$**:** $-\mathbf{y}^\top \mathbf{W}[s\mathbf{a}] + \mathbf{y}^\top[s\mathbf{Wa} + \mathbf{x}] = [\langle \mathbf{x}, \mathbf{y}\rangle].$

Next, we explain the security proof of this scheme by Abdalla et al. [4], which is somewhat different from the original proof by Agrawal et al. and roughly goes as follows. First, the form of the challenge ciphertext is changed from $\mathsf{ct} := ([s\mathbf{a}], [s\mathbf{Wa} + \mathbf{x}^\beta])$ to $\mathsf{ct} := ([s\mathbf{a} + s'\mathbf{b}], [\mathbf{W}(s\mathbf{a} + s'\mathbf{b}) + \mathbf{x}^\beta])$, where $s' \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, $\mathbf{b} := (1, 0)$, and $\beta \overset{\mathsf{U}}{\leftarrow} \{0, 1\}$. This change is computationally indistinguishable under the DDH assumption. At this point, we redefine $\mathbf{W}$ as

$$\mathbf{W} := \widetilde{\mathbf{W}} + u(\mathbf{x}^1 - \mathbf{x}^0)\mathbf{a}^{\perp^\top}, \tag{2.1}$$

where $u \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, $\widetilde{\mathbf{W}} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{m\times 2}$, and $\mathbf{a}^\perp := (1, -a)$, and note that $\mathbf{a}^{\perp^\top}\mathbf{b} = 1$. In fact, $\mathbf{x}^0$ and $\mathbf{x}^1$ may depend on $\widetilde{\mathbf{W}}$ because the information of $\widetilde{\mathbf{W}}$ is leaked to the adversary from the public key and queried secret keys. However, we can assume that $\mathbf{x}^0$ and $\mathbf{x}^1$ do not depend on $\widetilde{\mathbf{W}}$ (and formally we use complexity leveraging to argue that). Then, redefined $\mathbf{W}$ is also a random element in $\mathbb{Z}_p^{m\times 2}$ and we have

$$\mathbf{Wa} = \widetilde{\mathbf{W}}\mathbf{a}, \tag{2.2}$$

$$\mathbf{W}^\top \mathbf{y}_\ell = \widetilde{\mathbf{W}}^\top \mathbf{y}_\ell \quad (\ell \text{ is an index for the query number}), \tag{2.3}$$

$$\mathbf{W}(s\mathbf{a} + s'\mathbf{b}) + \mathbf{x}^\beta = \widetilde{\mathbf{W}}(s\mathbf{a} + s'\mathbf{b}) + us'(\mathbf{x}^1 - \mathbf{x}^0) + \mathbf{x}^\beta = \widetilde{\mathbf{W}}(s\mathbf{a} + s'\mathbf{b}) + (us' + \beta)(\mathbf{x}^1 - \mathbf{x}^0) + \mathbf{x}^0. \tag{2.4}$$

IPFE schemes						
scheme	$\|\mathsf{pk}\|$	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	sec. loss	assumption
ALS16 [6]	$m+1$	$2m$	$m+2$	$m+2$	$O(q_{\mathsf{ct}})$	DDH
AGRW17 [4]	$km+k^2+k$	$(k+1)m$	$m+k+1$	$m+k+1$	$O(q_{\mathsf{ct}})$	$\mathcal{D}_k$-MDDH
Ours	m^2+1	$2m^2$	$3m$	$3m$	$O(1)$	DDH
	$k^2m^2+k^2+k$	$(k^2+k)m^2$	$(k^2+k+1)m$	$(k^2+k+1)m$	$O(1)$	$\mathcal{D}_k$-MDDH

Table 1: Comparison of adaptively secure IPFE schemes in the multi-user and multi-challenge setting. The columns $|\mathsf{pk}|$ and $|\mathsf{ct}|$ refer to the number of group elements. The columns $|\mathsf{msk}|$ and $|\mathsf{sk}|$ refer to the number of $\mathbb{Z}_p$ elements. The number m refers to the vector length. The number q_{ct} refers to the total number of ciphertext queries by an adversary. Note that we omit the group description from $|\mathsf{pk}|$.

MIPFE schemes					
scheme	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	sec. loss	assumption
ACFGU18 [3]	$\{k^2+k+(k+2)m\}\mu$	$m+k+1$	$(m+k+1)\mu+1$	$O(q_{\mathsf{ct}})$	$\mathcal{D}_k$-MDDH
Ours	$(k^2m+km+1)m\mu$	$(k^2+k+1)m$	$(k^2+k+1)m\mu+1$	$O(1)$	$\mathcal{D}_k$-MDDH

Table 2: Comparison of MIPFE schemes based on a pairing-free group. The columns $|\mathsf{msk}|$ and $|\mathsf{sk}|$ refer to the number of $\mathbb{Z}_p$ elements. The column $|\mathsf{ct}|$ refers to the number of group elements. The number m refers to the vector length. The number μ refers to the number of slots. The number q_{ct} refers to the total number of ciphertext queries for all slots by an adversary.

function-hiding IPFE schemes					
scheme	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	sec. loss	assumption
DDM16 [18]	$8m^2+12m+28$	$4m+8$	$4m+8$	$O(q_{\mathsf{ct}}q_{\mathsf{sk}})$	SXDH
TAO16 [38]	$4m^2+18m+20$	$2m+5$	$2m+5$	$O(q_{\mathsf{ct}}+q_{\mathsf{sk}})$	XDLIN
Lin17 [34]	$(k+1)(4m+3k+1)$	$2m+2k+2$	$2m+2k+2$	$O(q_{\mathsf{ct}}+q_{\mathsf{sk}})$	$\mathcal{D}_k$-MDDH
Ours	$32m^2$	$10m$	$10m$	$O(1)$	SXDH
	$(4k^4+8k^3+12k^2+8k)m^2$	$(4k^2+4k+2)m$	$(4k^2+4k+2)m$	$O(1)$	$\mathcal{D}_k$-MDDH

Table 3: Comparison of fully function-hiding IPFE schemes in the standard model. Lin17 [34] refers to the scheme obtained by applying her paradigm to the IPFE scheme AGRW17 [4] (Appendix A). The column $|\mathsf{msk}|$ refers to the number of $\mathbb{Z}_p$ elements. The columns $|\mathsf{ct}|$ and $|\mathsf{sk}|$ refer to the number of group elements in G_1 and G_2 respectively. The number m refers to the vector length. The numbers q_{ct} and q_{sk} refer to the total numbers of ciphertext queries and secret key queries by an adversary respectively.

function-hiding MIPFE schemes			
scheme	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$
DOT18 [19]	$(2m+2k+1)^2\mu$	$2m+2k+1$	$(2m+2k+1)\mu$
ACFGU18 [3]	$\{(k+1)(4m+5k+1)+k\}\mu$	$2m+3k+2$	$(2m+3k+2)\mu(+\|G_T\|)$
Ours	$\{(k^4+2k^3+3k^2+2k)(2m+1)^2+m\}\mu$	$(2k^2+2k+1)(2m+1)$	$(2k^2+2k+1)(2m+1)\mu$
scheme	sec. loss	assumption	
DOT18 [19]	$O(q_{\mathsf{ct}}+q_{\mathsf{sk}})$	k-Lin	
ACFGU18 [3]	$O(q_{\mathsf{ct}}+\mu q_{\mathsf{sk}})$	$\mathcal{D}_k$-MDDH	
Ours	$O(1)$	$\mathcal{D}_k$-MDDH	

Table 4: Comparison of fully function-hiding MIPFE schemes. The column $|\mathsf{msk}|$ refers to the number of $\mathbb{Z}_p$ elements. The columns $|\mathsf{ct}|$ and $|\mathsf{sk}|$ refer to the number of group elements in G_1 and G_2 respectively. The number m refers to the vector length. The number μ refers to the number of slots. The numbers q_{ct} and q_{sk} refer to the total numbers of ciphertext queries for all slots and secret key queries by an adversary respectively.

In the indistinguishability-based security game, we impose a query condition on the adversary to avoid a trivial attack. That is, for all secret key queries, we have $\mathbf{x}^0\mathbf{y}_\ell = \mathbf{x}^1\mathbf{y}_\ell$. Eq. (2.3) follows from this condition. Finally, from Eq. (2.4), we can argue that the information of β is hidden from the adversary by the term us' unless $s' = 0$, because u is a fresh randomness from the viewpoint of the adversary. Thus, the scheme is secure under the DDH assumption. In the multi-challenge setting, however, this proof strategy needs a hybrid argument for each challenge and incurs the security loss of $O(q_{\mathrm{ct}})$, where q_{ct} is the number of the ciphertext challenges. Intuitively, this is because the matrix $\mathbf{W}$ is shared in all challenge ciphertexts and we cannot redefine $\mathbf{W}$ suitable for all challenge ciphertexts simultaneously in Eq. (2.1).

The first attempt to obtain a tight reduction is setting $\mathbf{W}$ in Eq. (2.1) as

$$u_1, \dots, u_L \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p, \quad \mathbf{W} := \widetilde{\mathbf{W}} + \sum_{\iota \in [L]} u_\iota \mathbf{x}_\iota {\mathbf{a}^{\perp}}^{\top},$$

where $L(\leq m)$ is the dimension of the space V spanned by $\mathbf{x}_j^1 - \mathbf{x}_j^0 \in \mathbb{Z}_p^m$ for all $j \in [q_{\mathrm{ct}}]$, and $\{\mathbf{x}_\iota\}_{\iota \in [L]}$ are a basis of V. In this case, Eq. (2.2) and Eq. (2.3) do not change and Eq. (2.4) becomes

$$\mathbf{W}(s_j\mathbf{a} + s_j'\mathbf{b}) + \mathbf{x}_j^\beta = \widetilde{\mathbf{W}}(s_j\mathbf{a} + s_j'\mathbf{b}) + s_j' \sum_{\iota \in [L]} u_\iota \mathbf{x}_\iota + \beta(\mathbf{x}_j^1 - \mathbf{x}_j^0) + \mathbf{x}_j^0,$$

where j is the index of challenge queries. If we can say that $\{[s_j' u_\iota]\}_{j \in [q_{\mathrm{ct}}], \iota \in [L]}$ are indistinguishable from $\{[r_{j,\iota}]\}_{j \in [q_{\mathrm{ct}}], \iota \in [L]}$, which are $q_{\mathrm{ct}}L$ random elements in G, we can conclude that the term $s_j' \sum_{\iota \in [L]} u_\iota \mathbf{x}_\iota$ hides the information of β. This is because $\mathbf{x}_j^1 - \mathbf{x}_j^0 \in V$ for all $j \in [q_{\mathrm{ct}}]$, and each $\sum_{\iota \in [L]} r_{j,\iota}\mathbf{x}_\iota$ is a completely random element in V. Fortunately, it is well known that $\{s_j' u_\iota\}_{j \in [q_{\mathrm{ct}}], \iota \in [L]}$ on the exponent forms a synthesizer [36], and they are computationally indistinguishable from $q_{\mathrm{ct}}L$ random group elements with the security loss being either q_{ct} or L. Thus, we can prove the security of the scheme by Agrawal et al. with the security loss of $O(m)$, which is independent from the adversaries' behavior.

However, the above proof contains two deficiencies. The first is that the security reduction is still not tight. The second is that the above strategy is useful against only selective adversaries. This is because the reduction algorithm needs to know about V to simulate each challenge ciphertext, but V depends on all challenge queries that the adversary makes. Thus, we have to overcome these two problems.

Toward tight security. The solution for the first problem (and partly for the second problem as a result) is to increase the column of the part $\mathbf{a}$, which allows us to embed more randomness into ciphertexts. That is, we modify the scheme as

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

$$a \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p, \quad \mathbf{W} \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p^{m \times 2m}, \quad \mathbf{a} := (a, 1), \quad \mathbf{A} := \mathbf{I}_m \otimes \mathbf{a} = \overbrace{\begin{pmatrix} \mathbf{a} & & & \\ & \mathbf{a} & & \\ & & \ddots & \\ & & & \mathbf{a} \end{pmatrix}}^{m \text{ vectors}} \in \mathbb{Z}_p^{2m \times m},$$

$$\mathsf{pk} := ([\mathbf{a}], [\mathbf{WA}]), \quad \mathsf{msk} := \mathbf{W}.$$

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$: $\mathbf{s} := (s_1, \dots, s_m) \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p^m, \quad \mathsf{ct} := ([\mathbf{As}], [\mathbf{WAs} + \mathbf{x}])$.

KeyGen(pk, msk, $\mathbf{y}$): sk := $(-\mathbf{W}^\top\mathbf{y}, \mathbf{y})$.

Dec(pk, ct, sk): $-\mathbf{y}^\top\mathbf{W}[\mathbf{A}\mathbf{s}] + \mathbf{y}^\top[\mathbf{W}\mathbf{A}\mathbf{s} + \mathbf{x}] = [\langle \mathbf{x}, \mathbf{y}\rangle]$.

The security proof goes as follows. First, the form of all challenge ciphertexts is changed to

$$\mathbf{B} := \mathbf{I}_m \otimes (1, 0) \in \mathbb{Z}_p^{2m\times m},\quad \mathbf{s}'_j := (s'_{j,1}, \dots, s'_{j,m}) \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_p^m,\quad \mathsf{ct} := ([\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j], [\mathbf{W}(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \mathbf{x}_j^\beta]). \tag{2.5}$$

The DDH problem is tightly reduced to the problem of distinguishing this change by the random self-reducibility. Next, we redefine $\mathbf{W}$ as

$$u \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_p,\quad \mathbf{W} := \widetilde{\mathbf{W}} + u \sum_{\iota\in[L]} \mathbf{x}_\iota \mathbf{a}_\iota^{\perp\top}, \tag{2.6}$$

where $\mathbf{a}_\iota^\perp \in \mathbb{Z}_p^{2m}$ is the ι-th column of $\mathbf{A}^\perp := \mathbf{I}_m \otimes \mathbf{a}^\perp$. Then, we have

$$\begin{aligned} &\mathbf{W}\mathbf{A} = \widetilde{\mathbf{W}}\mathbf{A},\\ &\mathbf{W}^\top\mathbf{y}_\ell = \widetilde{\mathbf{W}}^\top\mathbf{y}_\ell,\\ &\mathbf{W}(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \mathbf{x}_j^\beta = \widetilde{\mathbf{W}}(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + u \sum_{\iota\in[L]} s'_{j,\iota}\mathbf{x}_\iota + \beta(\mathbf{x}_j^1 - \mathbf{x}_j^0) + \mathbf{x}_j^0. \end{aligned} \tag{2.7}$$

In this case, we can see that $\{[us'_{j,\iota}]\}_{j\in[q_{\mathrm{ct}}],\iota\in[L]}$ are computationally indistinguishable from $\{[r_{j,\iota}]\}_{j\in[q_{\mathrm{ct}}],\iota\in[L]}$, which are $q_{\mathrm{ct}}L$ random elements in G, and this indistinguishability is tightly reduced to the DDH assumption by the random self-reducibility. Then, the information of β is completely hidden by the same argument as before in the selective security model.

Toward adaptive security. In this paragraph, we refer to the computational change from $\mathbf{A}\mathbf{s}_j$ to $\mathbf{A}\mathbf{s}_j+\mathbf{B}\mathbf{s}'_j$ as the first step and that from $\{[us'_{j,\iota}]\}_{j\in[q_{\mathrm{ct}}],\iota\in[L]}$ to $\{[r_{j,\iota}]\}_{j\in[q_{\mathrm{ct}}],\iota\in[L]}$ as the second step. The main obstacle to achieve the adaptive security is that the reduction algorithm needs to know about the space V before seeing all challenge queries in the second step. Our observation is that we do not need a random element in V to hide the information of β in each ciphertext. Let V_j be a space spanned by $\mathbf{x}_\iota^1 - \mathbf{x}_\iota^0 \in \mathbb{Z}_p^m$ for all $\iota \in [j]$. Then, a random element in V_j suffices to hide the information of β in the j-th ciphertext. Fortunately, the reduction algorithm knows about V_j when it simulates the j-th ciphertext because it already receives vectors that span V_j.

To do so, we modify the first step. In particular, we change the way of choosing $\mathbf{s}'_j$ in Eq. (2.5) as

$$s'_{j,1}, \dots, s'_{j,\phi(j)} \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_p,\quad \mathbf{s}'_j := (s'_{j,1}, \dots, s'_{j,\phi(j)}, 0^{m-\phi(j)}) \in \mathbb{Z}_p^m,$$

where $\phi(j) := \dim V_j$. Next, we modify the definition of $\mathbf{x}_\iota$ as $\mathbf{x}_\iota := \mathbf{x}_{\rho(\iota)}^1 - \mathbf{x}_{\rho(\iota)}^0 \in \mathbb{Z}_p^m$ for all $\iota \in [L]$, where $\rho(\iota) := \min \phi^{-1}(\iota)$. It is not difficult to confirm that $\{\mathbf{x}_\iota\}_{\iota\in[\phi(j)]}$ form a basis of V_j. Then, Eq. (2.7) is changed to

$$\mathbf{W}(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \mathbf{x}_j^\beta = \widetilde{\mathbf{W}}(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + u \sum_{\iota\in[\phi(j)]} s'_{j,\iota}\mathbf{x}_\iota + \beta(\mathbf{x}_j^1 - \mathbf{x}_j^0) + \mathbf{x}_j^0.$$

Observe that the reduction algorithm can compute $\mathbf{x}_\iota$ for $\iota \in [\phi(j)]$ when it simulates the j-th ciphertext. As explained in the previous paragraph, $\{[us'_{j,\iota}]\}_{j\in[q_{\mathrm{ct}}],\iota\in[\phi(j)]}$ are computationally indistinguishable from $\{[r_{j,\iota}]\}_{j\in[q_{\mathrm{ct}}],\iota\in[\phi(j)]}$, and the term $\sum_{\iota\in[\phi(j)]} r_{j,\iota}\mathbf{x}_\iota$ hides the information of β in the j-th ciphertext. Thus, we can achieve the adaptive security.

2.2 Conversion from Function-Hiding IPFE to Function-Hiding MIPFE

Similarly to previous MIPFE schemes, our conversion utilizes parallel execution of an underlying function-hiding IPFE scheme. The construction of our conversion can be seen as the combination of the non-function-hiding MIPFE scheme by Abdalla et al. [3] and the function-hiding MIPFE scheme by Datta et al. [19]. For simplicity, we consider the IPFE scheme over $\mathbb{Z}_n$ for some integer n, which means that the functionality of FE is inner product over $\mathbb{Z}_n$. Let m be a vector length and μ be a number of slots of the converted scheme, and $\mathsf{IPFE} := (\mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}')$ be an underlying weakly function-hiding IPFE scheme. Then, our conversion invokes Setup' with setting the vector length as $2m+1$ and generates μ master secret keys $\mathsf{msk}'_1, \ldots, \mathsf{msk}'_\mu$ (we omit public parameters here). In addition, it chooses μ random vectors $\mathbf{u}_1, \ldots, \mathbf{u}_\mu \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m$ and sets a master secret key of the converted scheme as $\mathsf{msk} := (\mathsf{msk}'_1, \ldots, \mathsf{msk}'_\mu, \mathbf{u}_1, \ldots, \mathbf{u}_\mu)$. To encrypt a vector $\mathbf{x}_i$ for the index i, it encrypts $\tilde{\mathbf{x}}_i := (\mathbf{x}_i + \mathbf{u}_i, 0^m, 1)$ as $\mathsf{ct}'_i \leftarrow \mathsf{Enc}'(\mathsf{msk}_i, \tilde{\mathbf{x}}_i)$ and outputs ct'_i. To generate a secret key for $\{\mathbf{y}_i\}_{i\in[\mu]}$, it first generates secret shares of $-\sum_{i\in[\mu]}\langle \mathbf{y}_i, \mathbf{u}_i\rangle$ as $r_1, \ldots, r_\mu \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n$ such that $\sum_{i\in[\mu]} r_i = -\sum_{i\in[\mu]}\langle \mathbf{y}_i, \mathbf{u}_i\rangle \pmod{n}$. These shares prevent the leakage of partial inner product values. Then, our conversion generates a secret key for $\tilde{\mathbf{y}}_i := (\mathbf{y}_i, 0^m, r_i)$ as $\mathsf{sk}'_i \leftarrow \mathsf{KeyGen}'(\mathsf{msk}'_i, \tilde{\mathbf{y}}_i)$ for all $i \in [\mu]$. Finally, it sets the secret key for converted scheme as $\mathsf{sk} := (\mathsf{sk}'_1, \ldots, \mathsf{sk}'_\mu)$. The decryption algorithm simply computes $\sum_{i\in[\mu]} \mathsf{Dec}'(\mathsf{ct}'_i, \mathsf{sk}'_i) \pmod{n}$. The correctness of the converted scheme is not difficult to confirm because $\sum_{i\in[\mu]}\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i\rangle = \sum_{i\in[\mu]}\langle \mathbf{x}_i, \mathbf{y}_i\rangle$.

Although our conversion is as simple as that by Abdalla et al. [3], the security proof needs a more ingenious technique. To see this, we briefly recall the proof strategy of their conversion and show that the naive application of their strategy to our conversion does not work. Here, we assume that the converted MIPFE scheme is weakly function-hiding, meaning that an adversary against the converted scheme has the following condition on the queries in the security game. Let $q_{\mathsf{ct},i}$ be the total number of ciphertext queries for index i and q_{sk} be the total number of secret key queries. Then, for all $(j_1, \ldots, j_\mu) \in [q_{\mathsf{ct},1}] \times \cdots \times [q_{\mathsf{ct},\mu}]$, and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\sum_{i\in[\mu]} \langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell}\rangle = \sum_{i\in[\mu]} \langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^1_{i,\ell}\rangle = \sum_{i\in[\mu]} \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^1_{i,\ell}\rangle. \tag{2.8}$$

The proof employs a series of games, and the goal is that the adversary does not obtain any information about a random bit β in the final game. The first step is to redefine $\mathbf{u}_i := \tilde{\mathbf{u}}_i + \mathbf{x}^0_{i,1} - \mathbf{x}^\beta_{i,1}$, where $\tilde{\mathbf{u}}_i \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n$. This information-theoretic change does not affect secret keys because $\sum_{i\in[\mu]}\langle \mathbf{x}^0_{i,1} - \mathbf{x}^\beta_{i,1}, \mathbf{y}^\beta_{i,\ell}\rangle = 0$ from Eq. (2.8). The second step is to change $\tilde{\mathbf{x}}_{i,j_i}$ from $(\mathbf{x}^\beta_{i,j_i} + \tilde{\mathbf{u}}_i + \mathbf{x}^0_{i,1} - \mathbf{x}^\beta_{i,1}, 0^m, 1)$ to $(\mathbf{x}^0_{i,j_i} + \tilde{\mathbf{u}}_i, 0^m, 1)$. This change is justified by the security of the underlying IPFE scheme because $\langle \mathbf{x}^\beta_{i,j_i} - \mathbf{x}^\beta_{i,1}, \mathbf{y}^\beta_{i,\ell}\rangle = \langle \mathbf{x}^0_{i,j_i} - \mathbf{x}^0_{i,1}, \mathbf{y}^\beta_{i,\ell}\rangle$ for all $i \in [\mu]$, which can be derived from Eq. (2.8). Finally, we want to change $\tilde{\mathbf{y}}_{i,\ell}$ from $(\mathbf{y}^\beta_{i,\ell}, 0^m, r_{i,\ell})$ to $(\mathbf{y}^0_{i,\ell}, 0^m, r'_{i,\ell})$ to hide the information of β. However, we cannot make this change in the adaptive setting. The reason is that the reduction algorithm needs to set $r'_{i,\ell} := r_{i,\ell} + \Delta_{i,\ell}$, where $\Delta_{i,\ell} := \langle \mathbf{x}^0_{i,j_i} + \mathbf{u}_i, \mathbf{y}^\beta_{i,\ell} - \mathbf{y}^0_{i,\ell}\rangle = \langle \mathbf{x}^0_{i,1} + \mathbf{u}_i, \mathbf{y}^\beta_{i,\ell} - \mathbf{y}^0_{i,\ell}\rangle$ (the second equality follows from Eq.(2.8)), to keep the inner product value when it simulates the ℓ-th secret key. If the adversary makes a secret key query before it makes the first ciphertext query for some index i, the reduction algorithm cannot simulate a secret key because it does not know the value $\langle \mathbf{x}^0_{i,1}, \mathbf{y}^\beta_{i,\ell} - \mathbf{y}^0_{i,\ell}\rangle$. Hence, this strategy does not work.

To circumvent this problem, we introduce another proof strategy. Recall that this problem occurs in the second step, where $\mathbf{y}^\beta_{i,\ell}$ is changed to $\mathbf{y}^0_{i,\ell}$, whereas the first step goes well, where $\mathbf{x}^\beta_{i,j_i}$ is changed to $\mathbf{x}^0_{i,j_i}$. Intuitively, our solution for this problem is to make both changes in one-shot in the same manner as the first step. That is, we do not take the intermediate step where the inner product values of queried vectors are $\sum_{i\in[\mu]}\langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^\beta_{i,\ell}\rangle$, and we change the replies such that the inner product values of queried vectors are directly changed from $\sum_{i\in[\mu]}\langle \mathbf{x}^\beta_{i,j_i}, \mathbf{y}^\beta_{i,\ell}\rangle$ to $\sum_{i\in[\mu]}\langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell}\rangle$. This means that our conversion allows us to directly achieve a fully function-hiding MIPFE scheme. This is possible if we prepare $2n+1$ dimensions for the underlying scheme and use the similar technique to that by Tomida et al. [38]. To do so, we want to create a situation where $\tilde{\mathbf{x}}_{i,j_i} := (\mathbf{x}^\beta_{i,j_i} + \tilde{\mathbf{u}}_i - \mathbf{x}^\beta_{i,1}, \mathbf{x}^0_{i,1}, 1)$ and $\tilde{\mathbf{y}}_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, \mathbf{y}^0_{i,\ell}, r'_{i,\ell})$. This is because if we have the above situation, we can change $\tilde{\mathbf{x}}_{i,j_i}$ to $(\tilde{\mathbf{u}}_i, \mathbf{x}^0_{i,j_i} - \mathbf{x}^0_{i,1} + \mathbf{x}^0_{i,1}, 1) = (\tilde{\mathbf{u}}_i, \mathbf{x}^0_{i,j_i}, 1)$ by the security of the underlying scheme and the relation $\langle \mathbf{x}^\beta_{i,j_i} - \mathbf{x}^\beta_{i,1}, \mathbf{y}^\beta_{i,\ell}\rangle = \langle \mathbf{x}^0_{i,j_i} - \mathbf{x}^0_{i,1}, \mathbf{y}^0_{i,\ell}\rangle$, which also can be derived from Eq.(2.8).

To reach the situation starting from the real game, however, we need one more trick. This is because the reduction algorithm needs to compute the value $\Delta_{i,\ell} := \langle \mathbf{x}^0_{i,1}, \mathbf{y}^0_{i,\ell}\rangle$ to adjust inner products with the term $r'_{i,\ell}$ when it simulates the ℓ-th secret key. Thus, the same problems as above occurs. To solve this problem, we take the intermediate step where $\tilde{\mathbf{x}}_{i,j_i} := (\mathbf{x}^\beta_{i,j_i} + \mathbf{u}_i, \mathbf{v}_i, 1)$ and $\tilde{\mathbf{y}}_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, \mathbf{y}^0_{i,\ell}, r_{i,\ell})$, where $\mathbf{v}_i \xleftarrow{\mathsf{U}} \mathbb{Z}^m_n$ is randomly chosen at the beginning of the game. This is possible because computing $\Delta_{i,\ell} := \langle \mathbf{v}_i, \mathbf{y}^0_{i,\ell}\rangle$ suffices for the reduction algorithm to reach the step. After the step, we redefine $\mathbf{u}_i := \tilde{\mathbf{u}}_i - \mathbf{x}^\beta_{i,1}$ and $\mathbf{v}_i := \tilde{\mathbf{v}}_i + \mathbf{x}^0_{i,1}$ where $\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \xleftarrow{\mathsf{U}} \mathbb{Z}^m_n$. This change is information-theoretic and we do not need to care about when the adversary makes the first ciphertext query. By these steps, our proof strategy goes well since there are no steps where reduction algorithms need to compute values related to $\mathbf{x}^0_{i,1}$ when it simulates secret keys.

The interesting points of our technique are to crucially utilize the blank space, namely the $n+1$ to $2n$-th dimensions, and directly construct a fully function-hiding MIPFE scheme from a weakly function-hiding IPFE scheme. This is in contrast to the function-hiding scheme in [3], where they first construct a weakly function-hiding MIPFE scheme, setting a vector length of an underlying IPFE scheme as almost n. Then, they convert it into a fully function-hiding scheme by doubling the vector length of the scheme.

3 Preliminary

3.1 Notation

For a natural number $n \in \mathbb{N}$, $\mathbb{Z}_n$ denotes a ring $\mathbb{Z}/n\mathbb{Z}$ and $[n]$ denotes a set $\{1,\dots,n\}$. For a set S, $s \xleftarrow{\mathsf{U}} S$ denotes that s is uniformly chosen from S. We treat vectors as column vectors. For a vector $\mathbf{x}$, $||\mathbf{x}||_\infty$ denotes its infinity norm. For vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$, $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$ denotes a vector generated by the vertical concatenation of these vectors. For matrices (including vectors) with the same number of rows $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$, $(\mathbf{A}_1||\mathbf{A}_2||\cdots||\mathbf{A}_n)$ denotes a matrix generated by the horizontal concatenation of these matrices. For a generator g_i of a cyclic group G_i of order p and $a \in \mathbb{Z}_p$, $[a]_i$ denotes g_i^a. Furthermore, for a matrix $\mathbf{A} := (a_{j,\ell})_{j,\ell}$ over $\mathbb{Z}_p$, $[\mathbf{A}]_i$ denotes a matrix over G_i whose (i,j) entry is $g_i^{a_{j,\ell}}$. For vectors $\mathbf{x} := (x_1,\dots,x_n)$ and $\mathbf{y} := (y_1,\dots,y_n) \in \mathbb{Z}^n_p$, let $e([\mathbf{x}]_1, [\mathbf{y}]_2) := e(g_1, g_2)^{\langle \mathbf{x},\mathbf{y}\rangle}$ be a function that computes the inner product on the exponent by $\prod_{i\in[n]} e([x_i]_1, [y_i]_2)$. A matrix $\mathbf{I}_n$ denotes the $n\times n$ identity matrix. A matrix $\mathbf{O}_{m\times n}$ denotes the $m \times n$ zero matrix. A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and denotes $f(\lambda) \leq \mathsf{negl}(\lambda)$. For families of distributions $X := \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y := \{Y_\lambda\}_{\lambda\in\mathbb{N}}$, $X \approx_c Y$ means that they are computationally indistinguishable.

3.2 Basic Tools and Assumption

Definition 3.1 (Cyclic Group). A description of a cyclic group $\mathbb{G}_{\mathrm{CG}}{:=}(p, G, g)$ consists of a prime p, a cyclic group G of order p, and a generator g. A cyclic group generator $\mathcal{G}_{\mathrm{CG}}(1^\lambda)$ takes a security parameter

1^λ and outputs a description of a cyclic group $\mathbb{G}_{\mathrm{CG}}$ with a λ-bit prime p.

Definition 3.2 (Bilinear Groups)**.** A description of bilinear groups $\mathbb{G}_{\mathrm{BG}}:=(p, G_1, G_2, G_T, g_1, g_2, e)$ consist of a prime p, cyclic groups G_1, G_2, G_T of order p, generators g_1 and g_2 of G_1 and G_2 respectively, and a bilinear map $e : G_1 \times G_2 \to G_T$, which has two properties.

- (Bilinearity): $\forall h_1 \in G_1, h_2 \in G_2, a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$.
- (Non-degeneracy): For generators g_1 and g_2, $g_T := e(g_1, g_2)$ is a generator of G_T.

A bilinear group generator $\mathcal{G}_{\mathrm{BG}}(1^\lambda)$ takes a security parameter 1^λ and outputs a description of bilinear groups $\mathbb{G}_{\mathrm{BG}}$ with a λ-bit prime p.

Definition 3.3 ($\mathcal{D}_k$-MDDH Assumption [21])**.** Let $\mathcal{D}_k$ be a matrix distribution over full rank matrices in $\mathbb{Z}_p^{(k+1)\times k}$. We can assume that, wlog, the first k rows of a matrix $\mathbf{A}$ chosen from $\mathcal{D}_k$ forms an invertible matrix. We consider the following distribution:

$$\mathbb{G}_{\mathrm{CG}} \leftarrow \mathcal{G}_{\mathrm{CG}}(1^\lambda), \;\; \mathbb{G}_{\mathrm{BG}} \leftarrow \mathcal{G}_{\mathrm{BG}}(1^\lambda),$$
$$\mathbf{A} \leftarrow \mathcal{D}_k, \;\; \mathbf{v} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k, \;\; \mathbf{t}_0 := \mathbf{A}\mathbf{v}, \;\; \mathbf{t}_1 \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k+1}.$$

We say that the $\mathcal{D}_k$-MDDH assumption holds with respect to $\mathcal{G}_{\mathrm{CG}}$ if, for any PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{A},\mathsf{CG}}^{\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{CG}}, [\mathbf{A}], [\mathbf{t}_0])] - \Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{CG}}, [\mathbf{A}], [\mathbf{t}_1])]| \le \mathsf{negl}(\lambda),$$

and with respect to $\mathcal{G}_{\mathrm{BG}}$ if, for any PPT adversary $\mathcal{A}$ and both $i \in \{1, 2\}$,

$$\mathsf{Adv}_{\mathcal{A},\mathsf{BG},i}^{\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{BG}}, [\mathbf{A}]_i, [\mathbf{t}_0]_i)] - \Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{BG}}, [\mathbf{A}]_i, [\mathbf{t}_1]_i)]| \le \mathsf{negl}(\lambda).$$

Random self-reducibility. By the random self-reducibility, we can obtain arbitrarily many instances of the $\mathcal{D}_k$-MDDH problem without additional security loss. For any $n \in \mathbb{N}$, we additionally define the following distribution:

$$\mathbf{V} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k\times n}, \;\; \mathbf{T}_0 := \mathbf{A}\mathbf{V}, \;\; \mathbf{T}_1 \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{(k+1)\times n}.$$

The advantages of $\mathcal{A}$ against n-fold $\mathcal{D}_k$-MDDH assumption with respect to $\mathcal{G}_{\mathrm{CG}}$ and $\mathcal{G}_{\mathrm{BG}}$ are defined as:

$$\mathsf{Adv}_{\mathcal{A},\mathsf{CG}}^{\mathsf{n}\text{-}\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{CG}}, [\mathbf{A}], [\mathbf{T}_0])] - \Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{CG}}, [\mathbf{A}], [\mathbf{T}_1])]|,$$
$$\mathsf{Adv}_{\mathcal{A},\mathsf{BG},i}^{\mathsf{n}\text{-}\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{BG}}, [\mathbf{A}]_i, [\mathbf{T}_0]_i)] - \Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathrm{BG}}, [\mathbf{A}]_i, [\mathbf{T}_1]_i)]|.$$

Then, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and both $i \in \{1, 2\}$, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_2$ and we have

$$\mathsf{Adv}_{\mathcal{A}_1,\mathsf{CG}}^{\mathsf{n}\text{-}\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) \le \mathsf{Adv}_{\mathcal{B}_1,\mathsf{CG}}^{\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) + 2^{-\Omega(\lambda)},$$
$$\mathsf{Adv}_{\mathcal{A}_2,\mathsf{BG},i}^{\mathsf{n}\text{-}\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) \le \mathsf{Adv}_{\mathcal{B}_2,\mathsf{BG},i}^{\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda) + 2^{-\Omega(\lambda)},$$
$$\mathsf{Time}(\mathcal{B}_j) \approx \mathsf{Time}(\mathcal{A}_j) + n\mathsf{poly}_j(\lambda) \;\; \text{for both } j \in \{1, 2\},$$

where $\mathsf{poly}_j(\lambda)$ is independent from $\mathsf{Time}(\mathcal{A}_j)$.

3.3 Definitions of Inner Product Functional Encryption

In this paper, we treat both single-input inner product functional encryption (IPFE) and multi-input IPFE. In both cases, the inner product functionality is defined over $\mathbb{Z}$ and its domain is limited depending on the infinity norms of the input vectors. We formally define the functionality called bonded-norm inner product.

Definition 3.4 (Bounded-Norm Inner Product over $\mathbb{Z}$)**.** This function family $\mathcal{F}$ consists of functions $f^{X,Y}_{\mathbf{y}_1,\ldots,\mathbf{y}_\mu}$: $\mathbb{Z}^m \times \cdots \times \mathbb{Z}^m \to \mathbb{Z}$ where $m, \mu, X, Y \in \mathbb{N}$, $\mathbf{y}_i \in \mathbb{Z}^m$ s.t. $||\mathbf{y}_i||_\infty \leq Y$. For all $(\mathbf{x}_1, \ldots, \mathbf{x}_\mu) \in (\mathbb{Z}^m)^\mu$ s.t. $\forall i \in [\mu]$, $||\mathbf{x}_i||_\infty \leq X$, we define the function as

$$f^{X,Y}_{\mathbf{y}_1,\ldots,\mathbf{y}_\mu}(\mathbf{x}_1, \ldots, \mathbf{x}_\mu) := \sum_{i \in [\mu]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle.$$

We call μ a number of slots. We refer to the function as single-input inner product when $\mu = 1$, and multi-input inner product when $\mu > 1$.

With respect to single-input IPFE, there are two types of IPFE: public-key IPFE and private-key IPFE. To achieve the function privacy, we need the private-key setting as defined below. Roughly speaking, this is because an adversary can learn the information of functions embedded in secret keys by decrypting ciphetexts generated by itself with the secret keys in the public-key setting.

Definition 3.5 (Public-Key Inner Product Functional Encryption)**.** Let $\mathcal{X} := \{X_\lambda\}_{\lambda \in \mathbb{N}}, \mathcal{Y} := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ be ensembles of norm-bounds. Public-key inner product functional encryption (Pub-IPFE) consists of five algorithms.

$\mathsf{Par}(1^\lambda)$**:** It takes a security parameter 1^λ and outputs a public parameter pp.

$\mathsf{Setup}(1^m, \mathsf{pp})$**:** It takes a vector length 1^m and pp and outputs a public key pk and a master secret key msk.

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** It takes pk and a vector $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ and outputs a ciphertext ct.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y})$**:** It takes pk, msk, and a vector $\mathbf{y} := (y_1, \ldots, y_m) \in \mathbb{Z}^m$ and outputs a secret key sk.

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk})$**:** It takes pk, ct and sk and outputs a decrypted value $d \in \mathbb{Z}$ or a symbol $\perp$.

Correctness. Pub-IPFE is *correct* if it satisfies the following condition. For any $\lambda, m \in \mathbb{N}$ and for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^m$ s.t. $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, we have

$$\Pr\left[d = \langle \mathbf{x}, \mathbf{y} \rangle \,\middle|\, \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{x}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y}) \\ d := \mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] = 1.$$

Security. Let $\mu \in \mathbb{N}$ be a natural number that represents the number of users. Pub-IPFE is *adaptively secure in the multi-user and multi-challenge setting* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ against Pub-IPFE defined as follows is negligible in λ for any constant $m, \mu \in \mathbb{N}$, and PPT

$\underline{O_{\mathsf{ct}}(\beta \in \{0,1\}, i \in [\mu], (\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^m)^2)}$	$\underline{O_{\mathsf{sk}}(i \in [\mu], \mathbf{y} \in \mathbb{Z}^m)}$
$\mathsf{ct}_i \overset{\mathsf{U}}{\longleftarrow} \mathsf{Enc}(\mathsf{pk}_i, \mathbf{x}^\beta)$ return ct_i	$\mathsf{sk}_i \overset{\mathsf{U}}{\longleftarrow} \mathsf{KeyGen}(\mathsf{pk}_i, \mathsf{msk}_i, \mathbf{y})$ return sk_i

Fig 1: The description of oracles in the security game for Pub-IPFE.

adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{Pub\text{-}IPFE}}_{\mathcal{A}}(\lambda) := \left| 2\Pr\left[\beta = \beta' \;\middle|\; \begin{array}{l} \beta \overset{\mathsf{U}}{\longleftarrow} \{0,1\}, \quad \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ \{(\mathsf{pk}_i, \mathsf{msk}_i)\}_{i \in [\mu]} \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \beta' \leftarrow \mathcal{A}^{O_{\mathsf{ct}}(\beta,\cdot,\cdot), O_{\mathsf{sk}}(\cdot,\cdot)}(1^\lambda, \{\mathsf{pk}_i\}_{i \in [\mu]}) \end{array} \right] - 1 \right|.$$

The description of the oracles O_{ct} and O_{sk} is presented in Fig 1. We refer to queries to O_{ct} and O_{sk} as a ciphertext query and a secret key query respectively. To avoid a trivial attack of $\mathcal{A}$, we have the following condition on $\mathcal{A}$'s queries. Let $q_{\mathsf{ct},i}$ and $q_{\mathsf{sk},i}$ be the total number of ciphertext queries and secret key queries for index i respectively. Then, for all $i \in [\mu]$, $j_i \in [q_{\mathsf{ct},i}]$, and $\ell_i \in [q_{\mathsf{sk},i}]$, we have

$$\langle \mathbf{x}^0_{i,j_i}, \mathbf{y}_{i,\ell_i} \rangle = \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}_{i,\ell_i} \rangle. \tag{3.1}$$

Definition 3.6 (Private-Key Inner Product Functional Encryption)**.** Let $\mathcal{X} := \{X_\lambda\}_{\lambda \in \mathbb{N}}, \mathcal{Y} := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ be ensembles of norm-bounds. Private-key inner product functional encryption (Priv-IPFE) consists of five algorithms.

$\mathsf{Par}(1^\lambda)$**:** It takes a security parameter 1^λ and outputs a public parameter pp.

$\mathsf{Setup}(1^m, \mathsf{pp})$**:** It takes a vector length 1^m and pp and outputs a master secret key msk.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$**:** It takes pp, msk, and a vector $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ and outputs a ciphertext ct.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathbf{y})$**:** It takes pp, msk, and a vector $\mathbf{y} := (y_1, \ldots, y_m) \in \mathbb{Z}^m$ and outputs a secret key sk.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk})$**:** It takes pp, ct and sk and outputs a decrypted value $d \in \mathbb{Z}$ or a symbol $\perp$.

Correctness. Priv-IPFE is *correct* if it satisfies the following condition. For any $\lambda, m \in \mathbb{N}$ and for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^m$ s.t. $||\mathbf{x}||_\infty \le X_\lambda$ and $||\mathbf{y}||_\infty \le Y_\lambda$, we have

$$\Pr\left[d = \langle \mathbf{x}, \mathbf{y} \rangle \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ \mathsf{msk} \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathbf{y}) \\ d := \mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] = 1.$$

Security. Let $\mu \in \mathbb{N}$ be a natural number that represents the number of users. Priv-IPFE is *fully function-hiding in the multi-user setting* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ against Priv-IPFE defined as follows is negligible in λ for any constant $m, \mu \in \mathbb{N}$ and any PPT adversary $\mathcal{A}$,

$$\begin{aligned} &\mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{A},\mathsf{f\text{-}fh}}(\lambda) \\ &:= \left| \Pr\left[\beta' = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ \{\mathsf{msk}_i\}_{i \in [\mu]} \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \beta' \leftarrow \mathcal{A}^{O_{\mathsf{ct}}(0,\cdot,\cdot), O_{\mathsf{sk}}(0,\cdot,\cdot)}(\mathsf{pp}) \end{array} \right] - \Pr\left[\beta' = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ \{\mathsf{msk}_i\}_{i \in [\mu]} \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \beta' \leftarrow \mathcal{A}^{O_{\mathsf{ct}}(1,\cdot,\cdot), O_{\mathsf{sk}}(1,\cdot,\cdot)}(\mathsf{pp}) \end{array} \right] \right|. \end{aligned}$$

$\underline{\mathcal{O}_{\mathsf{ct}}(\beta \in \{0,1\}, i \in [\mu], (\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^m)^2)}$	$\underline{\mathcal{O}_{\mathsf{sk}}(\beta \in \{0,1\}, i \in [\mu], (\mathbf{y}^0, \mathbf{y}^1) \in (\mathbb{Z}^m)^2)}$
$\mathsf{ct}_i \overset{\mathsf{U}}{\longleftarrow} \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}_i, \mathbf{x}^\beta)$	$\mathsf{sk}_i \overset{\mathsf{U}}{\longleftarrow} \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}_i, \mathbf{y}^\beta)$
return ct_i	return sk_i

Fig 2: The description of oracles in the security game for Priv-IPFE.

It is convenient for our paper to define the advantage on Priv-IPFE as above rather than the form like $|2\Pr[\beta = \beta'] - 1|$, and both formulations are equivalent. The description of the oracles $\mathcal{O}_{\mathsf{ct}}$ and $\mathcal{O}_{\mathsf{sk}}$ is presented in Fig 2. We refer to queries to $\mathcal{O}_{\mathsf{ct}}$ and $\mathcal{O}_{\mathsf{sk}}$ as a ciphertext query and a secret key query respectively. To avoid a trivial attack of $\mathcal{A}$, we have the following condition on $\mathcal{A}$'s queries. Let $q_{\mathsf{ct},i}$ and $q_{\mathsf{sk},i}$ be the total numbers of ciphertext queries and secret key queries for index i respectively. Then, for all $i \in [\mu]$, $j_i \in [q_{\mathsf{ct},i}]$, and $\ell_i \in [q_{\mathsf{sk},i}]$, we have

$$\langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell_i} \rangle = \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^1_{i,\ell_i} \rangle. \tag{3.2}$$

We say that Priv-IPFE is *weakly function-hiding in the multi-user setting* if it satisfies the above definition except that the query condition of $\mathcal{A}$ is more restricted as follows. That is, for all $i \in [\mu]$, $j_i \in [q_{\mathsf{ct},i}]$, and $\ell_i \in [q_{\mathsf{sk},i}]$, we have

$$\langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell_i} \rangle = \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^0_{i,\ell_i} \rangle = \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^1_{i,\ell_i} \rangle. \tag{3.3}$$

We denote the advantage of $\mathcal{A}$ in weakly function-hiding game in the multi-user setting by $\mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{A},\mathsf{w\text{-}fh}}(\lambda)$.

As pointed out by Abdalla et al. [4], public-key multi-input IPFE (MIPFE) is almost meaningless because it inherently leaks the same amount of information as parallel execution of single-input IPFE. Therefore, following them, we only consider private-key MIPFE in this paper.

Definition 3.7 (Multi-Input Inner Product Functional Encryption). Let $\mathcal{X} := \{X_\lambda\}_{\lambda \in \mathbb{N}}, \mathcal{Y} := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ be ensembles of norm-bound. Multi-input inner product functional encryption (MIPFE) consists of four algorithms.

$\mathsf{Setup}(1^\lambda, 1^m, 1^\mu)$**:** It takes a security parameter 1^λ, a vector length 1^m, and a number of slots 1^μ. Then, it outputs a public parameter pp and a master secret key msk.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x})$**:** It takes pp, msk, an index $i \in [\mu]$, and a vector $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ and outputs a ciphertext ct_i.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_i\}_{i \in [\mu]})$**:** It takes pp, msk, and vectors $\{\mathbf{y}_i := (y_{i,1}, \ldots, y_{i,m})\}_{i \in [\mu]} \in (\mathbb{Z}^m)^\mu$, and outputs a secret key sk.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\mu, \mathsf{sk})$**:** It takes $\mathsf{pp}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\mu$ and sk and outputs a decrypted value $d \in \mathbb{Z}$ or a symbol $\perp$.

Correctness. MIPFE is *correct* if it satisfies the following condition. For any $\lambda, m, \mu \in \mathbb{N}$ and for any $\{\mathbf{x}_i\}_{i \in [\mu]}, \{\mathbf{y}_i\}_{i \in [\mu]} \in (\mathbb{Z}^m)^\mu$ s.t. $\forall i, ||\mathbf{x}_i||_\infty \leq X_\lambda$ and $||\mathbf{y}_i||_\infty \leq Y_\lambda$, we have

$$\Pr\left[d = \sum_{i \in [\mu]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle \;\middle|\; \begin{array}{l} \mathsf{pp}, \mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda, 1^m, 1^\mu) \\ \mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x}_i) \quad \text{for all } i \in [\mu] \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_i\}_{i \in [\mu]}) \\ d := \mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] = 1.$$

$\underline{O_{\mathsf{ct}}(\beta \in \{0,1\}, i \in [\mu], (\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^m)^2)}$	$\underline{O_{\mathsf{sk}}(\beta \in \{0,1\}, (\{\mathbf{y}_i^0\}_{i\in[\mu]}, \{\mathbf{y}_i^1\}_{i\in[\mu]}) \in ((\mathbb{Z}^m)^\mu)^2)}$
$\mathsf{ct}_i \overset{\mathsf{U}}{\longleftarrow} \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x}^\beta)$ return ct_i	$\mathsf{sk} \overset{\mathsf{U}}{\longleftarrow} \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_i^\beta\}_{i\in[\mu]})$ return sk

Fig 3: The description of oracles in the security game for MIPFE.

Security. MIPFE is *fully function-hiding* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ against MIPFE defined as follows is negligible in λ for any constant $m, \mu \in \mathbb{N}$ and any PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{MIPFE}}_{\mathcal{A},\mathsf{f\text{-}fh}}(\lambda) := \left| 2\Pr\left[\beta = \beta' \;\middle|\; \begin{array}{l} \beta \overset{\mathsf{U}}{\longleftarrow} \{0,1\}, \\ (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^m, 1^\mu) \\ \beta' \leftarrow \mathcal{A}^{O_{\mathsf{ct}}(\beta,\cdot,\cdot), O_{\mathsf{sk}}(\beta,\cdot)}(\mathsf{pp}) \end{array} \right] - 1 \right|.$$

The description of the oracles O_{ct} and O_{sk} is presented in Fig 3. We refer to queries to O_{ct} and O_{sk} as a ciphertext query and a secret key query respectively. To avoid a trivial attack of $\mathcal{A}$, we have the following condition on $\mathcal{A}$'s queries. Let $q_{\mathsf{ct},i}$ be the total number of ciphertext queries for index i and q_{sk} be the total number of secret key queries. Then, for all $(j_1, \ldots, j_\mu) \in [q_{\mathsf{ct},1}] \times \cdots \times [q_{\mathsf{ct},\mu}]$, and $\ell \in [q_{\mathsf{sk}}]$,

$$\sum_{i\in[\mu]} \langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell} \rangle = \sum_{i\in[\mu]} \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^1_{i,\ell} \rangle. \tag{3.4}$$

In this paper, we assume that $q_{\mathsf{ct},i} \geq 1$ for all $i \in [\mu]$ and $q_{\mathsf{sk}} \geq 1$. Note that this condition can be easily removed by simply utilizing symmetric key encryption [4, 19].

We say that MIPFE is *adaptively secure* if it satisfies the above definition except that there is an additional query condition of $\mathcal{A}$. That is, for all $i \in [\mu]$ and $\ell \in [q_{\mathsf{sk}}]$,

$$\mathbf{y}^0_{i,\ell} = \mathbf{y}^1_{i,\ell}.$$

We denote the advantage of $\mathcal{A}$ in the adaptive-security game by $\mathsf{Adv}^{\mathsf{MIPFE}}_{\mathcal{A},\mathsf{ad}}(\lambda)$. This security definition captures only the message privacy of MIPFE schemes, i.e., the scheme is non-function-hiding. Note that this security definition of the adaptive security is identical to many-AD-IND security in [3, 4].

4 Tightly Secure (Multi-Input) Inner Product Functional Encryption

In this section, we present our tightly secure Pub-IPFE scheme and non-function-hiding MIPFE scheme, the latter is obtained by applying the conversion by Abdalla et al. [3] to our IPFE scheme.

4.1 Construction

Let $\mathcal{D}_k$ be a matrix distribution over full rank matrices in $\mathbb{Z}_p^{(k+1)\times k}$ and norm bounds X_λ and Y_λ be polynomials in λ.

$\mathsf{Par}(1^\lambda)$: It takes a security parameter 1^λ and outputs pp as follows.

$$\mathbb{G}_{\mathsf{CG}} \leftarrow \mathcal{G}_{\mathsf{CG}}(1^\lambda), \;\; \tilde{\mathbf{A}} \leftarrow \mathcal{D}_k, \;\; \mathsf{pp} := (\mathbb{G}_{\mathsf{CG}}, [\tilde{\mathbf{A}}])$$

$\mathsf{Setup}(1^m, \mathsf{pp})$**:** It takes a vector length 1^m and a public parameter pp. Then, it outputs a public key pk and a master secret key msk as follows.

$$\mathbf{W} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^{m \times k(k+1)m}, \quad \mathbf{A} := \overbrace{\begin{pmatrix} \tilde{\mathbf{A}} & & & \\ & \tilde{\mathbf{A}} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{A}} \end{pmatrix}}^{km \text{ matrices}} \in \mathbb{Z}_p^{k(k+1)m \times k^2 m}, \tag{4.1}$$

$$\mathsf{pk} := (\mathbb{G}_{\mathsf{CG}}, [\tilde{\mathbf{A}}], [\mathbf{WA}]), \quad \mathsf{msk} := \mathbf{W}.$$

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** It takes pk and $\mathbf{x} \in \mathbb{Z}^m$ and outputs a ciphertext ct as follows.

$$\mathbf{s} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^{k^2 m}, \quad \mathbf{c}_1 := \mathbf{As} \in \mathbb{Z}_p^{k(k+1)m}, \quad \mathbf{c}_2 := \mathbf{WAs} + \mathbf{x} \in \mathbb{Z}_p^m, \quad \mathsf{ct} := ([\mathbf{c}_1], [\mathbf{c}_2]).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y})$**:** It takes pp, msk, and $\mathbf{y} \in \mathbb{Z}^m$ and outputs a secret key sk as follows.

$$\mathbf{k}_1 := -\mathbf{W}^\top \mathbf{y} \in \mathbb{Z}_p^{k(k+1)m}, \quad \mathbf{k}_2 := \mathbf{y} \in \mathbb{Z}_p^m, \quad \mathsf{sk} := (\mathbf{k}_1, \mathbf{k}_2).$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk})$**:** It takes pk, ct, and sk. Then it computes $[d] := [\mathbf{k}_1^\top \mathbf{c}_1 + \mathbf{k}_2^\top \mathbf{c}_2]$ and searches for d exhaustively in the range of $-mX_\lambda Y_\lambda$ to $mX_\lambda Y_\lambda$. If such d is found, it outputs d. Otherwise, it outputs $\perp$.

Correctness. Observe that if ct is an encryption of $\mathbf{x}$ and sk is a secret key of $\mathbf{y}$,

$$d = -\mathbf{y}^\top \mathbf{WAs} + \mathbf{y}^\top \mathbf{WAs} + \mathbf{y}^\top \mathbf{x} = \langle \mathbf{x}, \mathbf{y} \rangle.$$

Therefore, if $||\mathbf{x}||_\infty \le X_\lambda$ and $||\mathbf{y}||_\infty \le Y_\lambda$, the output of the decryption algorithm is $d = \langle \mathbf{x}, \mathbf{y} \rangle$.

4.2 Security

Theorem 4.1. *Assume that the $\mathcal{D}_k$-MDDH assumption holds with respect to $\mathcal{G}_{\mathsf{CG}}$, then our Pub-IPFE scheme is adaptively secure in the multi-user and multi-challenge setting. More formally, let μ be a number of users, $q_{\mathsf{ct}} := \sum_{i \in [\mu]} q_{\mathsf{ct},i}$ be the total number of the ciphertext queries by $\mathcal{A}$, $q_{\mathsf{sk}} := \sum_{i \in [\mu]} q_{\mathsf{sk},i}$ be the total number of the secret key queries by $\mathcal{A}$, and m be a vector length. Then, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ for the $\mathcal{D}_k$-MDDH and we have*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Pub\text{-}IPFE}}(\lambda) \le 2\mathsf{Adv}_{\mathcal{B}_1,\mathsf{CG}}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}(\lambda) + 2\mathsf{Adv}_{\mathcal{B}_2,\mathsf{CG}}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}(\lambda) + 2^{-\Omega(\lambda)},$$
$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. We employ a series of games and evaluate the advantage of the adversary in each game. In this paper, we use the variable i to denote the index of users and j_i (resp. ℓ_i) to denote the index of ciphertext (resp. secret key) queries for user i. For example, a vector $\mathbf{s}$ in j_i-th ciphertext for user i will be denoted by $\mathbf{s}_{i,j_i}$. In the security proof, however, we change the forms of ciphertexts and secret keys for every user in the same way simultaneously. Thus, we do not need to specify users when we consider adversary's queries. For conciseness, we omit the index i from (i, j_i) and (i, ℓ_i), and just use j and ℓ to denote the indices of queries (but j and ℓ are implicitly associated with i).

Game 0: This game is the same as the real game. Then, for all $j \in [q_{\mathrm{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

$$\mathbf{s}_j \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k^2 m}, \quad \mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j, \quad \mathbf{c}_{j,2} := \mathbf{W}_i\mathbf{A}\mathbf{s}_j + \mathbf{x}_j^{\beta}.$$

Game 1: The reply for ciphertext queries is changed as follows. For $j \in [q_{\mathrm{ct},i}]$, we define $\mathbf{x}_j := \mathbf{x}_j^1 - \mathbf{x}_j^0 \in \mathbb{Z}_p^m$. Let $\phi_i : [q_{\mathrm{ct},i}] \to [m]$ be a map such that $\phi_i(j) := \mathsf{rank}(\mathbf{x}_1 \| \cdots \| \mathbf{x}_j)$. Then, for all $j \in [q_{\mathrm{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

$$\mathbf{b} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k+1} \backslash \mathsf{span}(\tilde{\mathbf{A}}), \quad \mathbf{B} := \overbrace{\begin{pmatrix} \mathbf{b} & & & \\ & \mathbf{b} & & \\ & & \ddots & \\ & & & \mathbf{b} \end{pmatrix}}^{km \text{ vectors}} \in \mathbb{Z}_p^{k(k+1)m \times km}, \tag{4.2}$$

$$\begin{aligned} &\tilde{\mathbf{s}}_{j,1}, \ldots, \tilde{\mathbf{s}}_{j,\phi_i(j)} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k, \quad \mathbf{s}'_j := (\tilde{\mathbf{s}}_{j,1}, \ldots, \tilde{\mathbf{s}}_{j,\phi_i(j)}, 0^{k(m-\phi_i(j))}) \in \mathbb{Z}_p^{km}, \\ &\mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j + \boxed{\mathbf{B}\mathbf{s}'_j}, \quad \mathbf{c}_{j,2} := \mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \boxed{\mathbf{B}\mathbf{s}'_j}) + \mathbf{x}_j^{\beta}. \end{aligned}$$

Game 2: The reply for ciphertext queries is changed as follows. Let $\rho_i : [\phi_i(q_{\mathrm{ct},i})] \to [q_{\mathrm{ct},i}]$ be a map such that $\rho_i(\iota) := \min \phi_i^{-1}(\iota)$. In other words, on an input ι, ρ_i returns the first query number j such that the rank of the matrix $(\mathbf{x}_1 \| \cdots \| \mathbf{x}_j)$ equals ι. Then, for all $j \in [q_{\mathrm{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

$$\begin{aligned} &\mathbf{u} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k, \\ &\mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j, \quad \mathbf{c}_{j,2} := \mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \mathbf{x}_j^{\beta} + \boxed{\sum_{\iota \in [\phi_i(j)]} \langle \mathbf{u}, \tilde{\mathbf{s}}_{j,\iota} \rangle \mathbf{x}_{\rho_i(\iota)}}. \end{aligned}$$

Note that $\tilde{\mathbf{s}}_{j,\iota}$ is defined in Game 1.

Game 3: The reply for ciphertext queries is changed as follows. For all $j \in [q_{\mathrm{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

$$\begin{aligned} &r_{j,1}, \ldots, r_{j,\phi_i(j)} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p, \\ &\mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j, \quad \mathbf{c}_{j,2} := \mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \mathbf{x}_j^{\beta} + \boxed{\sum_{\iota \in [\phi_i(j)]} r_{j,\iota} \mathbf{x}_{\rho_i(\iota)}}. \end{aligned}$$

Game 4: The reply for ciphertext queries is changed as follows. For all $j \in [q_{\mathrm{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

$$\begin{aligned} &r_{j,1}, \ldots, r_{j,\phi_i(j)} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p, \\ &\mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j, \quad \mathbf{c}_{j,2} := \mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \boxed{\mathbf{x}_j^0} + \sum_{\iota \in [\phi_i(j)]} r_{j,\iota} \mathbf{x}_{\rho_i(\iota)}. \end{aligned}$$

Thanks to Lemma 4.1 to Lemma 4.5, Theorem 4.1 holds. □

In the following, we denote the event that $\mathcal{A}$'s output is equal to β, i.e., $\beta = \beta'$, in Game ι by E_ι.

Lemma 4.1. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_1$ for the $\mathcal{D}_k$-MDDH s.t.*

$$|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + 2^{-\Omega(\lambda)},$$
$$\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. We describe a PPT adversary $\mathcal{B}_1$ that solves a $\mathcal{D}_k$-MDDH problem using $\mathcal{A}$ internally. $\mathcal{B}_1$ takes an n-$\mathcal{D}_k$-MDDH problem $(\mathbb{G}_{\mathsf{CG}}, [\tilde{\mathbf{A}}], [\mathbf{T}_\delta])$ with $n := kmq_{\mathsf{ct}}$, where $\delta \in \{0, 1\}$. Note that the number $n = kmq_{\mathsf{ct}}$ corresponds to the maximum possible instance usage of $\mathcal{B}_1$. Therefore, the number of instances that $\mathcal{B}_1$ utilizes depends on $\mathcal{A}$'s behavior and $\mathcal{B}_1$ does not utilize all instances necessarily. $\mathcal{B}_1$ generates random matrices $\mathbf{W}_1, \ldots, \mathbf{W}_\mu \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{m \times k(k+1)m}$ and sets $\mathsf{pk}_i := (\mathbb{G}_{\mathsf{CG}}, [\tilde{\mathbf{A}}], [\mathbf{W}_i\mathbf{A}])$ for all $i \in [\mu]$, where $\mathbf{A}$ is defined in the same way as Eq. (4.1). Then, $\mathcal{B}_1$ inputs $\{\mathsf{pk}_i\}_{i\in[\mu]}$ to $\mathcal{A}$. Because $\mathcal{B}_1$ generates $\mathsf{msk}_i := \mathbf{W}_i$ for all i by itself, it can easily simulate $\mathcal{O}_{\mathsf{sk}}$. Thus, the remaining task is simulating $\mathcal{O}_{\mathsf{ct}}$.

First, $\mathcal{B}_1$ selects a bit $\beta \overset{\mathsf{U}}{\leftarrow} \{0, 1\}$. Let $\mathbf{t}_{\delta,\iota} \in \mathbb{Z}_p^{k+1}$ be the ι-th column of $\mathbf{T}_\delta$. When $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{ct}}$ on $(i, (\mathbf{x}_{j,0}, \mathbf{x}_{j,1}))$ as the j-th query for user i, $\mathcal{B}_1$ computes a reply as follows:

$$\mathbf{s}_{j,k\phi_i(j)+1}, \ldots, \mathbf{s}_{j,km} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k,$$
$$\mathbf{c}_{j,1} := (\mathbf{t}_{\delta,km(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+1}, \ldots, \mathbf{t}_{\delta,km(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+k\phi(j)}, \tilde{\mathbf{A}}\mathbf{s}_{j,k\phi_i(j)+1}, \ldots, \tilde{\mathbf{A}}\mathbf{s}_{j,km}) \in \mathbb{Z}_p^{k(k+1)m},$$
$$\mathbf{c}_{j,2} := \mathbf{W}_i\mathbf{c}_{j,1} + \mathbf{x}_j^\beta \in \mathbb{Z}_p^m,$$
$$\mathsf{ct}_j := ([\mathbf{c}_{j,1}], [\mathbf{c}_{j,2}]).$$

We check that $\mathcal{B}_1$ correctly simulates $\mathcal{O}_{\mathsf{ct}}$. Recall that the columns of $\tilde{\mathbf{A}}$ and $\mathbf{b}$ are linearly independent and form a basis of $\mathbb{Z}_p^m$. Thus, we can rewrite $\mathbf{c}_{j,1}$ as:

$$\mathbf{s}_{j,1}, \ldots, \mathbf{s}_{j,km} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k, \quad s'_{j,1}, \ldots, s'_{j,k\phi_i(j)} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p,$$
$$\mathbf{c}_{j,1} = (\tilde{\mathbf{A}}\mathbf{s}_{j,1} + \delta s'_{j,1}\mathbf{b}, \ldots, \tilde{\mathbf{A}}\mathbf{s}_{j,k\phi_i(j)} + \delta s'_{j,k\phi_i(j)}\mathbf{b}, \tilde{\mathbf{A}}\mathbf{s}_{j,k\phi_i(j)+1}, \ldots, \tilde{\mathbf{A}}\mathbf{s}_{j,km})$$
$$= \mathbf{A}\mathbf{s}_j + \delta\mathbf{B}\mathbf{s}'_j,$$

where $\mathbf{s}_j := (\mathbf{s}_{j,1}, \ldots, \mathbf{s}_{j,km})$ and $\mathbf{s}'_j := (s_{j,1}, \ldots, s_{j,k\phi_i(j)}, 0^{k(m-\phi_i(j))})$. Then, if $\delta = 0$, $\mathcal{A}$'s view corresponds to Game 0 and otherwise, it corresponds to Game 1. Finally, $\mathcal{B}_1$ outputs the truth value of $(\beta = \beta')$ where β' is the output of $\mathcal{A}$. This proves Lemma 4.1. □

Lemma 4.2. *For any PPT adversary $\mathcal{A}$, we have*

$$\Pr[\mathsf{E}_1] = \Pr[\mathsf{E}_2].$$

Proof. Lemma 4.2 follows from Claim 4.1 and Claim 4.2. To prove Lemma 4.2, we use a kind of complexity leveraging argument. In the following, we randomly choose vectors independently from the security game as $\{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathsf{ct},i}]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^m$. The purpose is to assure that $\tilde{\mathbf{x}}_j$ is independent from $\widetilde{\mathbf{W}}_i$ in Eq. (4.4). □

Claim 4.1. *For any PPT adversary $\mathcal{A}$ and both $\iota \in \{1, 2\}$, we have*

$$\Pr[\mathsf{E}_\iota] = \Pr[\mathsf{E}_\iota | \{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathsf{ct},i}]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^m, \forall i, j,\ \tilde{\mathbf{x}}_j = \mathbf{x}_j \pmod{p}].$$

Proof. Vectors $\{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathrm{ct},i}]}$ are chosen independently from $\mathcal{A}$'s view. Then, the event $[\forall i,j,\ \tilde{\mathbf{x}}_j = \mathbf{x}_j \pmod{p}]$ does not affect $\mathcal{A}$'s behavior. ◻

Claim 4.2. *For any PPT adversary $\mathcal{A}$, we have*

$$\begin{aligned}&\Pr[\mathsf{E}_1 | \{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathrm{ct},i}]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^m, \forall i,j,\ \tilde{\mathbf{x}}_j = \mathbf{x}_j \pmod{p}]\\ =&\Pr[\mathsf{E}_2 | \{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathrm{ct},i}]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^m, \forall i,j,\ \tilde{\mathbf{x}}_j = \mathbf{x}_j \pmod{p}].\end{aligned}$$

Proof. We denote the ι-th column of the matrix $\mathbf{B}$ by $\mathbf{b}_\iota$ for $\iota \in [km]$, where $\mathbf{B}$ is defined in Eq. (4.2). We define that $\mathbf{B}^* := \left((\mathbf{A}||\mathbf{B})^{-1}\right)^\top \in \mathbb{Z}_p^{k(k+1)m\times k(k+1)m}$ and denote the $(k^2m+\iota)$-th column of $\mathbf{B}^*$ by $\mathbf{b}_\iota^*$ for $\iota \in [km]$. Then the following equations hold:

$$\mathbf{b}_\iota^{*\top}\mathbf{A} = \mathbf{0}^\top, \quad \mathbf{b}_\iota^{*\top}\mathbf{b}_{\iota'} = \begin{cases}1 & (\iota = \iota')\\ 0 & (\iota \neq \iota')\end{cases} \quad \text{for all } \iota,\iota' \in [km]. \tag{4.3}$$

Next, we redefine $\mathbf{W}_i$ as

$$\begin{aligned}&\mathbf{u} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k,\ \ \widetilde{\mathbf{W}}_i \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{m\times k(k+1)m},\\ &\mathbf{W}_i := \widetilde{\mathbf{W}}_i + \sum_{\iota\in[\phi_i(q_{\mathrm{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top\left(\mathbf{b}^*_{k(\iota-1)+1}||\ldots||\mathbf{b}^*_{k(\iota-1)+k}\right)^\top.\end{aligned} \tag{4.4}$$

Observe that $\mathbf{W}_i$ is identically distributed to the original one, i.e., $\mathbf{W}_i \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{m\times k(k+1)m}$. This is because $\tilde{\mathbf{x}}_j$ is determined independently from $\widetilde{\mathbf{W}}_i$. Under the condition such that $\forall i,j,\ \tilde{\mathbf{x}}_j = \mathbf{x}_j \pmod{p}$, we have

(In the public key)

$$\mathbf{W}_i\mathbf{A} = \widetilde{\mathbf{W}}_i\mathbf{A} \quad \text{for all } i \in [\mu], \tag{4.5}$$

(In the secret keys)

$$\mathbf{W}_i^\top\mathbf{y}_\ell = \widetilde{\mathbf{W}}_i^\top\mathbf{y}_\ell \quad \text{for all } i\in[\mu] \text{ and } \ell \in [q_{\mathsf{sk},i}], \tag{4.6}$$

(In the challenge ciphertexts)

$$\begin{aligned}\mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) &= \left(\widetilde{\mathbf{W}}_i + \sum_{\iota\in[\phi_i(q_{\mathrm{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top\left(\mathbf{b}^*_{k(\iota-1)+1}||\ldots||\mathbf{b}^*_{k(\iota-1)+k}\right)^\top\right)(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j)\\ &= \widetilde{\mathbf{W}}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \sum_{\iota\in[\phi_i(q_{\mathrm{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top\left(\mathbf{O}_{k\times k(\iota-1)}||\mathbf{I}_k||\mathbf{O}_{k\times k(m-\iota)}\right)\mathbf{s}'_j\\ &= \widetilde{\mathbf{W}}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \sum_{\iota\in[\phi_i(j)]} \langle\mathbf{u},\tilde{\mathbf{s}}_{j,\iota}\rangle\tilde{\mathbf{x}}_{\rho_i(\iota)} \quad \text{for all } i\in[\mu] \text{ and } j \in [q_{\mathrm{ct},i}].\end{aligned} \tag{4.7}$$

Here, Eq. (4.5) and Eq. (4.7) follow from Eq. (4.3), and Eq. (4.6) follows from Eq. (3.1). Then, from Eq. (4.5), Eq. (4.6), and Eq. (4.7), $\mathcal{A}$'s views in Game 1 and Game 2 are identical if $\forall i,j,\ \tilde{\mathbf{x}}_j = \mathbf{x}_j \pmod{p}$. Then, Claim 4.2 holds. ◻

Lemma 4.3. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_2$ for the $\mathcal{D}_k$-MDDH s.t.*

$$\begin{aligned}&|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_3]| \leq \mathsf{Adv}_{\mathcal{B}_2}^{\mathcal{D}_k\text{-MDDH}}(\lambda) + 2^{-\Omega(\lambda)},\\ &\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathrm{ct}} + q_{\mathrm{sk}})\mathsf{poly}(\lambda, m),\end{aligned}$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. First, we prove the following claim.

Claim 4.3. *We consider the following distribution for any $n \in \mathbb{N}$:*

$$\mathbb{G}_{\mathsf{CG}} \leftarrow \mathcal{G}_{\mathsf{CG}}(1^\lambda),\ \ \mathbf{S} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k\times n},\ \ \mathbf{u} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k,\ \ \mathbf{t}_0 := \mathbf{S}^\top \mathbf{u},\ \ \mathbf{t}_1 \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^n.$$

Then, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ and we have

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Problem}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathsf{CG}}, [\mathbf{S}], [\mathbf{t}_0])] - \Pr[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathsf{CG}}, [\mathbf{S}], [\mathbf{t}_1])]| \leq \mathsf{Adv}_{\mathcal{B},\mathsf{CG}}^{\mathsf{n}\text{-}\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}(\lambda),$$
$$\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + n\mathsf{poly}(\lambda),$$

where $\mathsf{poly}(\lambda)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. For a matrix $\mathbf{A}$ in the n-$\mathcal{D}_k$-MDDH problem, we can define that $\mathbf{A} := \begin{pmatrix}\mathbf{A}_0 \\ \mathbf{a}_1^\top\end{pmatrix}$, where $\mathbf{A}_0 \in \mathsf{GL}_k(\mathbb{Z}_p)$ and $\mathbf{a}_1 \in \mathbb{Z}_p^k$. Then, we can rewrite an instance of n-$\mathcal{D}_k$-MDDH problem as

$$\mathbf{S} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k\times n},\ \ \tilde{\mathbf{t}}_0 := \mathbf{S}^\top (\mathbf{A}_0^{-1})^\top \mathbf{a}_1,\ \ \tilde{\mathbf{t}}_1 \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^n,$$
$$\left(\mathbb{G}_{\mathsf{CG}},\ \ \mathbf{A} := \begin{pmatrix}\mathbf{A}_0 \\ \mathbf{a}_1^\top\end{pmatrix},\ \ \mathbf{T}_0 := \begin{pmatrix}\mathbf{S} \\ \tilde{\mathbf{t}}_0^\top\end{pmatrix}\ \text{ or }\ \mathbf{T}_1 := \begin{pmatrix}\mathbf{S} \\ \tilde{\mathbf{t}}_1^\top\end{pmatrix}\right).$$

$\mathcal{B}$ chooses $\mathbf{r} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k$, sets $\mathbf{t}_\beta := \mathbf{S}^\top \mathbf{r} + \tilde{\mathbf{t}}_\beta$ for $\beta \in \{0,1\}$, and inputs $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{S}], [\mathbf{t}_\beta])$ to $\mathcal{A}$. Observe that $\mathbf{S}$ and $\mathbf{t}_\beta$ defined above are identically distributed to those defined in Claim 4.3. ◻

We describe a PPT adversary $\mathcal{B}_2$ that solves a problem defined in Claim 4.3 using $\mathcal{A}$ internally. $\mathcal{B}_2$ takes an instance $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{S}], [\mathbf{t}_\delta])$ with $n := mq_{\mathsf{ct}}$, where $\delta \in \{0,1\}$. Note that the number $n = mq_{\mathsf{ct}}$ corresponds to the maximum possible instance usage of $\mathcal{B}_2$. Therefore, the number of instances that $\mathcal{B}_2$ utilizes depends on $\mathcal{A}$'s behavior, and $\mathcal{B}_2$ does not utilize all instances necessarily. $\mathcal{B}_2$ chooses $\tilde{\mathbf{A}} \leftarrow \mathcal{D}_k$ and sets $\mathsf{pp} := (\mathbb{G}_{\mathsf{CG}}, [\tilde{\mathbf{A}}])$. $\mathcal{B}_2$ generates key pairs as $(\mathsf{pk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^m, \mathsf{pp})$ for all $i \in [\mu]$. Then, $\mathcal{B}_2$ inputs $\{\mathsf{pk}_i\}_{i\in[\mu]}$ to $\mathcal{A}$. Because $\mathcal{B}_2$ generates msk_i for all i by itself, it can easily simulate $\mathcal{O}_{\mathsf{sk}}$. Then, the remaining task is simulating $\mathcal{O}_{\mathsf{ct}}$.

First, $\mathcal{B}_2$ selects a bit $\beta \overset{\mathsf{U}}{\leftarrow} \{0,1\}$. Let $\tilde{\mathbf{s}}_\iota \in \mathbb{Z}_p^k$ be the ι-th column of $\mathbf{S} \in \mathbb{Z}_p^{k\times n}$ and $t_{\delta,\iota} \in \mathbb{Z}_p$ be the ι-th element of $\mathbf{t}_\delta \in \mathbb{Z}_p^n$. When $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{ct}}$ on $(i, (\mathbf{x}_{j,0}, \mathbf{x}_{j,1}))$ as the j-th query for user i, $\mathcal{B}_2$ computes a reply as follows:

$$\mathbf{s}_j \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k^2 m},\ \ \mathbf{s}'_j := \left(\tilde{\mathbf{s}}_{m(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+1}, \ldots, \tilde{\mathbf{s}}_{m(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+\phi_i(j)}, 0^{k(m-\phi_i(j))}\right) \in \mathbb{Z}_p^{km},$$
$$\mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j,\ \ \mathbf{c}_{j,2} := \mathbf{W}_i \mathbf{c}_{j,1} + \mathbf{x}_j^\beta + \sum_{\kappa\in[\phi_i(j)]} t_{\delta, m(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+\kappa}\mathbf{x}_{\rho_i(\kappa)},$$
$$\mathsf{ct}_j := ([\mathbf{c}_{j,1}], [\mathbf{c}_{j,2}]),$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined as Eq.(4.1) and Eq.(4.2). Recall that $t_{0,m(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+\kappa} = \langle \mathbf{u}, \tilde{\mathbf{s}}_{m(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+\kappa}\rangle$ and $t_{1,m(\sum_{\iota\in[i-1]} q_{\mathsf{ct},\iota}+j-1)+\kappa}$ is a random element in $\mathbb{Z}_p$ for $\kappa \in [\phi_i(j)]$. Then, $\mathcal{A}$'s view corresponds to Game 2 if $\delta = 0$, and it corresponds to Game 3 otherwise. Finally, $\mathcal{B}_2$ outputs the truth value of $(\beta = \beta')$ where β' is the output of $\mathcal{A}$. This proves Lemma 4.3. ◻

Lemma 4.4. *For any PPT adversary $\mathcal{A}$, we have*

$$\Pr[\mathsf{E}_3] = \Pr[\mathsf{E}_4].$$

Proof. For any $i \in [\mu]$ and $j \in [q_{\mathsf{ct},i}]$, we can see that the term $\sum_{\iota \in [\phi_i(j)]} r_{j,\iota} \mathbf{x}_{\rho_i(\iota)}$ is a completely random element in $\mathsf{span}(\{\mathbf{x}_{\rho_i(\iota)}\}_{\iota \in [\phi_i(j)]})$. From the definition of the maps ϕ_i and ρ_i, $\mathbf{x}_j \in \mathsf{span}(\{\mathbf{x}_{\rho_i(\iota)}\}_{\iota \in [\phi_i(j)]})$. Therefore, we have

$$\begin{aligned} \mathbf{x}_j^\beta + \sum_{\iota \in [\phi_i(j)]} r_{j,\iota} \mathbf{x}_{\rho_i(\iota)} &= \beta \mathbf{x}_j + \mathbf{x}_j^0 + \sum_{\iota \in [\phi_i(j)]} r_{j,\iota} \mathbf{x}_{\rho_i(\iota)} \\ &\equiv \mathbf{x}_j^0 + \sum_{\iota \in [\phi_i(j)]} r_{j,\iota} \mathbf{x}_{\rho_i(\iota)} \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}]. \end{aligned}$$

In the above equation, the relation $\equiv$ indicates that LHS and RHS are identically distributed. Thus, $\mathcal{A}$'s views in Game 3 and Game 4 are identical. □

Lemma 4.5. *For any PPT adversary $\mathcal{A}$, we have*

$$\Pr[\mathsf{E}_4] = 1/2.$$

Lemma 4.5 is trivial because $\mathcal{A}$ does not obtain any information about β in Game 4.

4.3 Application to Multi-Input Inner Product Functional Encryption

We can obtain an adaptively secure MIPFE scheme whose security is tightly reduced to the $\mathcal{D}_k$ - MDDH assumption by applying the generic conversion by Abdalla et al. [3] to our scheme. Let Pub-IPFE be a Pub-IPFE scheme that is adaptively secure in the multi-user and multi-challenge setting. It is not difficult to see that the security of the MIPFE scheme obtained by applying the conversion to Pub-IPFE is reduced to that of Pub-IPFE with the security loss being 1. For the completeness, we describe their conversion in a slightly modified way so that it is sufficient for our purpose.

Property. Let Pub-IPFE := (Par, Setup, Enc, KeyGen, Dec) be a Pub-IPFE scheme (Definition 3.5). In their conversion, we require that Pub-IPFE has the following properties.

1. Pub-IPFE is adaptively secure in the multi-challenge and multi-user setting.
2. A public parameter pp defines an order n, a group G of order n with group law $\circ$, and an encoding function $E : \mathbb{Z}_n \to G$.
3. A decryption algorithm Dec correctly works even if it takes pp instead of pk. Moreover, the decryption algorithm Dec can be divided into the two algorithms Dec_1 and Dec_2 with the following properties. For any $\lambda, m \in \mathbb{N}$, any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^m$, and any $z \in \mathbb{Z}_n$ such that $|z| \le mX_\lambda Y_\lambda$, we have

$$\Pr\left[d = E(\langle \mathbf{x}, \mathbf{y} \rangle \mod n) \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{x}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y}) \\ d := \mathsf{Dec}_1(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] = 1, \quad \mathsf{Dec}_2(\mathsf{pp}, E(z)) = z.$$

4. For any $a, b \in \mathbb{Z}_n$, we have $E(a) \circ E(b) = E(a + b)$.
5. Given pp and any $z \in \mathbb{Z}_n$, one can efficiently compute $E(-z)$.

Conversion by Abdalla et al. [3]. Let $\mathsf{Pub\text{-}IPFE} := (\mathsf{Par}', \mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}' := (\mathsf{Dec}'_1, \mathsf{Dec}'_2))$ be a Pub-IPFE scheme with the property defined above. Let $\mathsf{MIPFE} := (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ be a converted MIPFE scheme. Let $X_\lambda := X'_\lambda/\mu$ be a norm bound of MIPFE, where X'_λ is a norm bound of Priv-IPFE.

$\mathsf{Setup}(1^\lambda, 1^m, 1^\mu)$**:** It takes a security parameter 1^λ, a vector length 1^m, and a number of slots 1^μ. Then, it outputs a public parameter pp and a master secret key msk as follows.

$$\begin{aligned}&\mathsf{pp}' \leftarrow \mathsf{Par}'(1^\lambda), \quad \{\mathsf{pk}'_i, \mathsf{msk}'_i\}_{i\in[\mu]} \leftarrow \mathsf{Setup}'(1^m, \mathsf{pp}'), \quad \{\mathbf{u}_i\}_{i\in[\mu]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m,\\ &\mathsf{pp} := \mathsf{pp}', \quad \mathsf{msk} := (\{\mathsf{pk}'_i, \mathsf{msk}'_i\}_{i\in[\mu]}, \{\mathbf{u}_i\}_{i\in[\mu]}).\end{aligned}$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x})$**:** It takes pp, msk, $i \in [\mu]$ and $\mathbf{x} \in \mathbb{Z}^m$ and outputs a ciphertext ct_i as follows.

$$\tilde{\mathbf{x}} := \mathbf{x} + \mathbf{u}_i \in \mathbb{Z}_n^m, \quad \mathsf{ct}'_i \leftarrow \mathsf{Enc}'(\mathsf{pk}'_i, \tilde{\mathbf{x}}), \quad \mathsf{ct}_i := \mathsf{ct}'_i.$$

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_i\}_{i\in[\mu]})$**:** It takes pp, msk, and $\{\mathbf{y}_i\}_{i\in[\mu]} \in \mathbb{Z}^m$ and outputs a secret key sk as follows.

$$\begin{aligned}&\tilde{\mathbf{y}}_i := \mathbf{y}_i \in \mathbb{Z}_n^m, \quad \mathsf{sk}'_i \leftarrow \mathsf{KeyGen}'(\mathsf{pk}'_i, \mathsf{msk}'_i, \tilde{\mathbf{y}}_i) \ \text{ for all } i \in [\mu],\\ &z := \sum_{i\in[\mu]} \langle \mathbf{y}_i, \mathbf{u}_i \rangle \in \mathbb{Z}_n, \quad \mathsf{sk} := (\{\mathsf{sk}'_i\}_{i\in[\mu]}, z).\end{aligned}$$

$\mathsf{Dec}(\mathsf{pp}, \{\mathsf{ct}_i\}_{i\in[\mu]}, \mathsf{sk})$**:** It takes pp, $\{\mathsf{ct}_i\}_{i\in[\mu]}$, and sk. Then, it computes decryption value d as follows.

$$d_i := \mathsf{Dec}'_1(\mathsf{pp}', \mathsf{ct}'_i, \mathsf{sk}'_i) \in \mathbb{G} \ \text{ for all } i \in [\mu], \quad d := \mathsf{Dec}'_2(\mathsf{pp}, d_1 \circ \cdots \circ d_\mu \circ E(-z)).$$

By the conversion, we obtain the following corollary.

Corollary 4.1. *Let MIPFE be the MIPFE scheme obtained by applying the conversion in [3] to our Pub-IPFE scheme. Then MIPFE is adaptively secure. More formally, let μ be a number of slots, $q_{\mathsf{ct}} := \sum_{i\in[\mu]} q_{\mathsf{ct},i}$ be the total number of the ciphertext queries by $\mathcal{A}$, q_{sk} be the total number of the secret key queries by $\mathcal{A}$, and m be a vector length. Then, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ for the $\mathcal{D}_k$-MDDH and we have*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{MIPFE}}_{\mathcal{A},\mathsf{ad}}(\lambda) \leq 2\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B}_1}(\lambda) + 2\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)},\\ &\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + \mu q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),\end{aligned}$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

5 Function-Hiding Inner Product Functional Encryption

Lin proposed a simple framework that allows us to construct a function-hiding IPFE scheme from a public key IPFE scheme [34]. We can apply her framework to our scheme and obtain a tightly function-hiding IPFE scheme in the multi-user setting. Informally, her framework is as follows.

First, we can see that a ciphertext and a secret key in our IPFE scheme consist of vectors, and decryption involves inner product of these vectors. That is, a ciphertext of a vector $\mathbf{x}$ corresponds to a vector $\mathbf{c}_{\mathsf{in}} := (\mathbf{c}_{\mathsf{in},1}, \mathbf{c}_{\mathsf{in},2}) := (\mathbf{A}\mathbf{s}, \mathbf{W}\mathbf{A}\mathbf{s} + \mathbf{x}) \in \mathbb{Z}_p^{(k^2+k+1)m}$ and a secret key of a vector $\mathbf{y}$ corresponds to a vector $\mathbf{k}_{\mathsf{in}} :=$

$(\mathbf{k}_{\mathsf{in},1}, \mathbf{k}_{\mathsf{in},2}) := (-\mathbf{W}^\top \mathbf{y}, \mathbf{y}) \in \mathbb{Z}_p^{(k^2+k+1)m}$. Decryption just computes $\langle \mathbf{c}_{\mathsf{in}}, \mathbf{k}_{\mathsf{in}} \rangle$. We call the scheme described above an inner scheme.

To ensure the confidentiality of secret keys, we "encrypt" secret keys in the same way as ciphertexts in our IPFE scheme. That is, a secret key of the function-hiding IPFE scheme is generated as $\mathsf{sk} := (\mathbf{c}_{\mathsf{out},1}, \mathbf{c}_{\mathsf{out},2}) := \left(\mathbf{Dr} \in \mathbb{Z}_p^{k(k+1)(k^2+k+1)m}, \mathbf{VDr} + \mathbf{k}_{\mathsf{in}} \in \mathbb{Z}_p^{(k^2+k+1)m}\right)$, where $\mathbf{V}$, $\mathbf{D}$, and $\mathbf{r}$ correspond to $\mathbf{W}$, $\mathbf{A}$, and $\mathbf{s}$ respectively in our scheme presented in Section 4.1. We call the scheme utilized to encrypt secret keys an outer scheme. We also need to transform ciphertexts to make them compatible with sk, which can be done by "generating a secret key" of $\mathbf{c}_{\mathsf{in}}$ in the outer scheme. That is, we define a ciphertext of the function-hiding IPFE scheme as $\mathsf{ct} := (\mathbf{k}_{\mathsf{out},1}, \mathbf{k}_{\mathsf{out},2}) := \left(-\mathbf{V}^\top \mathbf{c}_{\mathsf{in}} \in \mathbb{Z}_p^{k(k+1)(k^2+k+1)m}, \mathbf{c}_{\mathsf{in}} \in \mathbb{Z}_p^{(k^2+k+1)m}\right)$. Observe that $\langle \mathsf{ct}, \mathsf{sk} \rangle = \langle \mathbf{c}_{\mathsf{in}}, \mathbf{k}_{\mathsf{in}} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$.

To achieve the security, of course we need to encode both ct and sk on the exponent of group elements. We employ bilinear groups that allow us to compute inner product over the group elements, which is necessary for decryption. Then, the confidentiality of ciphertexts is assured by the inner scheme and that of secret keys is assured by the outer scheme.

5.1 Actual Scheme and Optimization

As described above, if we directly apply Lin's framework to our scheme, the first components of a ciphertext and a secret key will consist of $k(k+1)(k^2+k+1)m$ group elements. Recall the reason we need $k(k+1)m$ group elements in the first components of a ciphertext and a secret key in the original scheme. That is, the maximum dimension of the space spanned by the vectors $\mathbf{x}_j = \mathbf{x}_j^1 - \mathbf{x}_j^0$ is m, and this fact directly affects the number of group elements in the first components. Because the vector length handled in the outer scheme is $(k^2+k+1)m$, the first components seem to require $k(k+1)(k^2+k+1)m$ group elements. However, observe that the maximum dimension of the space spanned by the vectors $\mathbf{k}_{\mathsf{out},\ell} := \mathbf{k}^1_{\mathsf{out},\ell} - \mathbf{k}^0_{\mathsf{out},\ell} := (-\mathbf{W}^\top \mathbf{y}^1_\ell, \mathbf{y}^1_\ell) - (-\mathbf{W}^\top \mathbf{y}^0_\ell, \mathbf{y}^0_\ell)$ for all $\ell \in [q_{\mathsf{sk}}]$ is m, not $(k^2+k+1)m$. Hence, we can reduce the number of group elements in the first components to $k(k+1)m$, and the resulting scheme is given as follows.

Let $\mathcal{D}_k$ be a matrix distribution over full rank matrices in $\mathbb{Z}_p^{(k+1)\times k}$ and norm bounds X_λ and Y_λ be polynomials in λ.

$\mathsf{Par}(1^\lambda)$**:** It takes a security parameter 1^λ and outputs pp as follows.

$$\mathbb{G}_{\mathsf{BG}} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \ \ \tilde{\mathbf{A}}, \tilde{\mathbf{D}} \leftarrow \mathcal{D}_k, \ \ \mathsf{pp} := (\mathbb{G}_{\mathsf{BG}}, [\tilde{\mathbf{A}}]_1, [\tilde{\mathbf{D}}]_2).$$

$\mathsf{Setup}(1^m, \mathsf{pp})$**:** It takes a vector length 1^m and a public parameter pp. Then, it outputs a master secret key msk as follows.

$$\mathbf{W} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^{m \times k(k+1)m}, \ \ \mathbf{V} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^{(k^2+k+1)m \times k(k+1)m}, \ \ \mathsf{msk} := (\mathbf{W}, \mathbf{V}).$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$**:** It takes pp, msk, and $\mathbf{x} \in \mathbb{Z}^m$ and outputs a ciphertext ct as follows.

$$\mathbf{A} := \overbrace{\begin{pmatrix} \tilde{\mathbf{A}} & & & \\ & \tilde{\mathbf{A}} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{A}} \end{pmatrix}}^{km \text{ matrices}} \in \mathbb{Z}_p^{k(k+1)m \times k^2 m}, \ \ \mathbf{s} \xleftarrow{\mathsf{U}} \mathbb{Z}_p^{k^2 m}, \ \ \mathbf{c}_{\mathsf{in}} := (\mathbf{As}, \mathbf{WAs} + \mathbf{x}) \in \mathbb{Z}_p^{(k^2+k+1)m},$$

$$\mathbf{k}_{\mathsf{out},1} := -\mathbf{V}^\top \mathbf{c}_{\mathsf{in}} \in \mathbb{Z}_p^{k(k+1)m}, \ \ \mathbf{k}_{\mathsf{out},2} := \mathbf{c}_{\mathsf{in}}, \ \ \mathsf{ct} := ([\mathbf{k}_{\mathsf{out},1}]_1, [\mathbf{k}_{\mathsf{out},2}]_1).$$

KeyGen(pp, msk, y): It takes pp, msk, and $\mathbf{y} \in \mathbb{Z}^m$ and outputs a secret key sk as follows.

$$\mathbf{D} := \overbrace{\begin{pmatrix} \tilde{\mathbf{D}} & & & \\ & \tilde{\mathbf{D}} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{D}} \end{pmatrix}}^{km \text{ matrices}} \in \mathbb{Z}_p^{k(k+1)m \times k^2 m}, \quad \mathbf{r} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{k^2 m}, \quad \mathbf{k}_{\mathsf{in}} := (-\mathbf{W}^\top \mathbf{y}, \mathbf{y}) \in \mathbb{Z}_p^{(k^2+k+1)m},$$

$$\mathbf{c}_{\mathsf{out},1} := \mathbf{D}\mathbf{r} \in \mathbb{Z}_p^{k(k+1)m}, \quad \mathbf{c}_{\mathsf{out},2} := \mathbf{V}\mathbf{D}\mathbf{r} + \mathbf{k}_{\mathsf{in}} \in \mathbb{Z}_p^{(k^2+k+1)m}, \quad \mathsf{sk} := ([\mathbf{c}_{\mathsf{out},1}]_2, [\mathbf{c}_{\mathsf{out},2}]_2).$$

Dec(pp, ct, sk): It takes pp, ct, and sk. Then it computes $[d]_T := e([\mathbf{k}_{\mathsf{out},1}]_1, [\mathbf{c}_{\mathsf{out},1}]_2) e([\mathbf{k}_{\mathsf{out},2}]_1, [\mathbf{c}_{\mathsf{out},2}]_2)$ and searches for d exhaustively in the range of $-mX_\lambda Y_\lambda$ to $mX_\lambda Y_\lambda$. If such d is found, it outputs d. Otherwise, it outputs $\perp$.

Correctness. Observe that if ct is an encryption of $\mathbf{x}$ and sk is a secret key of $\mathbf{y}$,

$$d = -\mathbf{c}_{\mathsf{in}}^\top \mathbf{V}\mathbf{D}\mathbf{r} + \mathbf{c}_{\mathsf{in}}^\top \mathbf{V}\mathbf{D}\mathbf{r} + \mathbf{c}_{\mathsf{in}}^\top \mathbf{k}_{\mathsf{in}} = \langle \mathbf{c}_{\mathsf{in}}, \mathbf{k}_{\mathsf{in}} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle.$$

Therefore, if $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, the output of the decryption algorithm is $d = \langle \mathbf{x}, \mathbf{y} \rangle$.

5.2 Security

Theorem 5.1. *Assume that the $\mathcal{D}_k$-MDDH assumption holds with respect to $\mathcal{G}_{\mathrm{BG}}$, then our Priv-IPFE scheme is weakly function-hiding in the multi-user setting. More formally, let μ be a number of users, $q_{\mathsf{ct}} := \sum_{i \in [\mu]} q_{\mathsf{ct},i}$ be the total number of the ciphertext queries by $\mathcal{A}$, $q_{\mathsf{sk}} := \sum_{i \in [\mu]} q_{\mathsf{sk},i}$ be the total number of the secret key queries by $\mathcal{A}$, and m be a vector length. Then, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exist PPT adversaries $\mathcal{B}_1, \ldots, \mathcal{B}_4$ for the $\mathcal{D}_k$-MDDH, and we have*

$$\mathsf{Adv}_{\mathcal{A},\mathsf{w\text{-}fh}}^{\mathsf{Priv\text{-}IPFE}}(\lambda) \leq 2 \sum_{\iota \in \{1,2\}} \mathsf{Adv}_{\mathcal{B}_\iota,\mathsf{BG},1}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}(\lambda) + 2 \sum_{\iota \in \{3,4\}} \mathsf{Adv}_{\mathcal{B}_\iota,\mathsf{BG},2}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}(\lambda) + 2^{-\Omega(\lambda)},$$

$$\max_{\iota \in [4]} \{\mathsf{Time}(\mathcal{B}_\iota)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + q_{\mathsf{sk}}) \mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Theorem 5.1 follows from Theorem 4.1 and Lin's observation [34]. That is, the following relations hold:

$$\left\{ \{\mathsf{ct}_j^0\}_{j \in [q_{\mathsf{ct},i}]}, \{\mathsf{sk}_\ell^0\}_{\ell \in [q_{\mathsf{sk},i}]} \right\}_{i \in [\mu]} \approx_c \left\{ \{\mathsf{ct}_j^1\}_{j \in [q_{\mathsf{ct},i}]}, \{\mathsf{sk}_\ell^0\}_{\ell \in [q_{\mathsf{sk},i}]} \right\}_{i \in [\mu]} \approx_c \left\{ \{\mathsf{ct}_j^1\}_{j \in [q_{\mathsf{ct},i}]}, \{\mathsf{sk}_\ell^1\}_{\ell \in [q_{\mathsf{sk},i}]} \right\}_{i \in [\mu]}.$$

The first indistinguishability follows from the security of the inner scheme and Eq. (3.3), and the second indistinguishability follows from the security of the outer scheme and Eq. (3.3). More precisely, we use the relations $\langle \mathbf{x}_{i,j_i}^0, \mathbf{y}_{i,\ell_i}^0 \rangle = \langle \mathbf{x}_{i,j_i}^1, \mathbf{y}_{i,\ell_i}^0 \rangle$ for the inner scheme and $\langle \mathbf{c}_{\mathsf{in},i,j_i}^1, \mathbf{k}_{\mathsf{in},i,\ell_i}^0 \rangle = \langle \mathbf{c}_{\mathsf{in},i,j_i}^1, \mathbf{k}_{\mathsf{in},i,\ell_i}^1 \rangle$ for the outer scheme. Both relations can be derived from Eq. (3.3). Note that because our scheme is adaptively secure, the above relations hold even if ciphertexts and secret keys are queried by an adversary adaptively.

Remark 5.1. Although the above scheme is weakly function-hiding in the multi-user setting, we can easily convert it into one that is fully function-hiding in the multi-user setting by the conversion proposed by Lin and Vaikuntanathan [35]. The conversion is very simple and works by only doubling vector lengths. When encrypting $\mathbf{x} \in \mathbb{Z}^m$, we just encrypt $(\mathbf{x}, 0^m)$ in the original scheme. Key generation is also done in the same way. In addition, this conversion is tight. That is, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ and we have

$$\begin{aligned} &\mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{A},\mathsf{f\text{-}fh}}(\lambda) \leq \sum_{\iota \in [3]} \mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{B}_\iota,\mathsf{w\text{-}fh}}(\lambda), \\ &\max_{\iota \in [3]}\{\mathsf{Time}(\mathcal{B}_\iota)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + q_{\mathsf{sk}})\mathsf{poly}(\lambda, m), \end{aligned}$$

where $\mathsf{poly}(\lambda, m)$ is independent from $\mathsf{Time}(\mathcal{A})$.

6 From Single to Multi-Input Function-Hiding Inner Product Functional Encryption

In this section, we present a generic conversion from weakly function-hiding single-input IPFE to fully function-hiding multi-input IPFE. Because all known function-hiding single-input IPFE schemes are based on bilinear groups, we design the conversion to be compatible with group based schemes. As in [3], however, we believe that our conversion is so generic that we can easily modify it to be suitable to schemes based on other primitives if constructed.

6.1 Conversion

Property. Let $\mathsf{Priv\text{-}IPFE} := (\mathsf{Par}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ be a Priv-IPFE scheme (Definition 3.6). In our conversion, we require that an underlying scheme has the following properties.

1. Priv-IPFE is weakly function-hiding in the multi-user setting.
2. A public parameter pp defines an order n, a group G of order n with group law $\circ$, and an encoding function $E : \mathbb{Z}_n \to G$.
3. A decryption algorithm Dec can be divided into the two algorithms Dec_1 and Dec_2 with the following properties. For any $\lambda, m \in \mathbb{N}$, any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^m$, and any $z \in \mathbb{Z}_n$ such that $|z| \leq mX_\lambda Y_\lambda$, we have

$$\Pr\left[d = E(\langle \mathbf{x}, \mathbf{y} \rangle \mod n) \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ \mathsf{msk} \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathbf{y}) \\ d := \mathsf{Dec}_1(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] = 1, \quad \mathsf{Dec}_2(\mathsf{pp}, E(z)) = z.$$

4. For any $a, b \in \mathbb{Z}_n$, we have $E(a) \circ E(b) = E(a + b)$.

Conversion. Let $\mathsf{Priv\text{-}IPFE} := \big(\mathsf{Par}', \mathsf{Setup}', \mathsf{Enc}', \mathsf{KeyGen}', \mathsf{Dec}' := (\mathsf{Dec}'_1, \mathsf{Dec}'_2)\big)$ be a Priv-IPFE scheme with the property defined above. Let $\mathsf{MIPFE} := (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ be a converted MIPFE scheme. Let $X_\lambda := X'_\lambda/\mu$ be a norm bound of MIPFE, where X'_λ is a norm bound of Priv-IPFE. Our conversion is performed as follows.

$\mathsf{Setup}(1^\lambda, 1^m, 1^\mu)$**:** It takes a security parameter 1^λ, a vector length 1^m, and a number of slots 1^μ. Then, it outputs a public parameter pp and a master secret key msk as follows.

$$\begin{aligned}&\mathsf{pp}' \leftarrow \mathsf{Par}'(1^\lambda),\ \ \{\mathsf{msk}'_i\}_{i\in[\mu]} \leftarrow \mathsf{Setup}'(1^{2m+1}, \mathsf{pp}'),\ \ \{\mathbf{u}_i\}_{i\in[\mu]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m,\\ &\mathsf{pp} := \mathsf{pp}',\ \ \mathsf{msk} := (\{\mathsf{msk}'_i\}_{i\in[\mu]}, \{\mathbf{u}_i\}_{i\in[\mu]}).\end{aligned}$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x})$**:** It takes pp, msk, $i \in [\mu]$ and $\mathbf{x} \in \mathbb{Z}^m$ and outputs a ciphertext ct_i as follows.

$$\tilde{\mathbf{x}} := (\mathbf{x} + \mathbf{u}_i, 0^m, 1) \in \mathbb{Z}_n^{2m+1},\ \ \mathsf{ct}'_i \leftarrow \mathsf{Enc}'(\mathsf{pp}', \mathsf{msk}'_i, \tilde{\mathbf{x}}),\ \ \mathsf{ct}_i := \mathsf{ct}'_i.$$

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \{\mathbf{y}_i\}_{i\in[\mu]})$**:** It takes pp, msk, and $\{\mathbf{y}_i\}_{i\in[\mu]} \in \mathbb{Z}^m$ and outputs a secret key sk as follows.

$$\begin{aligned}&\{r_i\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n,\ \ r_\mu := -\left(\sum_{i\in[\mu-1]} r_i + \sum_{i\in[\mu]} \langle \mathbf{y}_i, \mathbf{u}_i\rangle\right) \in \mathbb{Z}_n,\\ &\tilde{\mathbf{y}}_i := (\mathbf{y}_i, 0^m, r_i) \in \mathbb{Z}_n^{2m+1},\ \ \mathsf{sk}'_i \leftarrow \mathsf{KeyGen}'(\mathsf{pp}', \mathsf{msk}'_i, \tilde{\mathbf{y}}_i)\ \text{ for all } i \in [\mu],\\ &\mathsf{sk} := \{\mathsf{sk}'_i\}_{i\in[\mu]}.\end{aligned}$$

$\mathsf{Dec}(\mathsf{pp}, \{\mathsf{ct}_i\}_{i\in[\mu]}, \mathsf{sk})$**:** It takes pp, $\{\mathsf{ct}_i\}_{i\in[\mu]}$, and sk. Then, it computes decryption value d as follows.

$$d_i := \mathsf{Dec}'_1(\mathsf{pp}', \mathsf{ct}'_i, \mathsf{sk}'_i) \in \mathbb{G}\ \text{ for all } i \in [\mu],\ \ d := \mathsf{Dec}'_2(\mathsf{pp}', d_1 \circ \cdots \circ d_\mu).$$

Correctness. From property 3, we have

$$d_i = E(\langle \mathbf{x}_i + \mathbf{u}_i, \mathbf{y}_i\rangle + r_i \mod n).$$

From property 4, we have

$$d_1 \circ \cdots \circ d_\mu = E\left(\sum_{i\in[\mu]} (\langle \mathbf{x}_i + \mathbf{u}_i, \mathbf{y}_i\rangle + r_i) \mod n\right) = E\left(\sum_{i\in[\mu]} \langle \mathbf{x}_i, \mathbf{y}_i\rangle \mod n\right).$$

Then, from property 3 and the correctness of Priv-IPFE, we have $d := \mathsf{Dec}'_2(d_1 \circ \cdots \circ d_\mu) = \sum_{i\in[\mu]}\langle \mathbf{x}_i, \mathbf{y}_i\rangle$.

Remark 6.1. Typically, we define Priv-IPFE as consisting of four algorithms (Setup, Enc, KeyGen, Dec) and Setup outputs pp and msk when we consider Priv-IPFE in the single-user setting. To apply our conversion to such a Priv-IPFE scheme, just setting $\mathsf{pp} := \mathsf{pp}'_1, \ldots, \mathsf{pp}'_\mu$ suffices in the setup algorithm. In the security proof, however, we need a hybrid argument for each slot similarly to [3]. Thus, the security reduction will not become tight.

6.2 Security

Theorem 6.1. *Let Priv-IPFE be a Priv-IPFE scheme that satisfies the properties described above. Then converted scheme, MIPFE, is a fully function-hiding MIPFE scheme. More formally, let μ be a number of slots, $q_{\mathsf{ct}} := \sum_{i\in[\mu]} q_{\mathsf{ct},i}$ be the total number of the ciphertext queries by $\mathcal{A}$, q_{sk} be the total number of the secret key queries*

game	$\tilde{\mathbf{x}}_{i,j}$ in ct	$\tilde{\mathbf{y}}_{i,\ell}$ in sk	$-\sum r_{i,\ell}$	justification
0 (real)	$(\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, 0^m, 1)$	$(\mathbf{y}^{\beta}_{i,\ell}, 0^m, r_{i,\ell})$	$\sum\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle$	-
1	$(\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \boxed{\mathbf{v}_i}, 1)$	$(\mathbf{y}^{\beta}_{i,\ell}, 0^m, r_{i,\ell})$	$\sum\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle$	w-fh
2	$(\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \mathbf{v}_i, 1)$	$(\mathbf{y}^{\beta}_{i,\ell}, \boxed{\mathbf{y}^{0}_{i,\ell}}, r_{i,\ell})$	$\boxed{\sum(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle)}$	w-fh
3	$(\mathbf{x}^{\beta}_{i,j} \boxed{-\mathbf{x}^{\beta}_{i,1}} + \mathbf{u}_i, \boxed{\mathbf{x}^{0}_{i,1}} + \mathbf{v}_i, 1)$	$(\mathbf{y}^{\beta}_{i,\ell}, \mathbf{y}^{0}_{i,\ell}, r_{i,\ell})$	$\sum(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle)$	info.
4	$(\mathbf{u}_i, \boxed{\mathbf{x}^{0}_{i,j}} + \mathbf{v}_i, 1)$	$(\mathbf{y}^{\beta}_{i,\ell}, \mathbf{y}^{0}_{i,\ell}, r_{i,\ell})$	$\sum(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle)$	w-fh
5 (final)	$(\mathbf{u}_i, \mathbf{x}^{0}_{i,j} + \mathbf{v}_i, 1)$	$(\boxed{0^m}, \mathbf{y}^{0}_{i,\ell}, r_{i,\ell})$	$\boxed{\sum\langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle}$	w-fh

Table 5: Overview of the game change. In justification, w-fh stands for the weakly function-hiding security of Priv-IPFE and info. stands for an information-theoretic change.

by $\mathcal{A}$, and m be a vector length. Then, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_2$ for Priv-IPFE and we have

$$\mathsf{Adv}^{\mathsf{MIPFE}}_{\mathcal{A},\mathsf{f\text{-}fh}}(\lambda) \le 2\sum_{\iota\in[2]} \mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{B}_\iota,\mathsf{w\text{-}fh}}(\lambda),$$

$$\max_{\iota\in[2]}\{\mathsf{Time}(\mathcal{B}_\iota)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + \mu q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. We employ a series of games and evaluate the advantage of the adversary in each game. For ease of exposition, we first consider six games: Games 0 to 5, and show that the each transition of games is justified by the security of the underlying scheme (or an information-theoretical argument). Then, we explain that the transition from Game 0 to 2 and that from Game 3 to 5 can be done in one-shot. We summarize forms of ciphertexts and secret keys in each game in Table 5. A formal description of each game is given as follows. Similarly to in Section 4.2, we omit index i from index j_i and just denote it by j.

Game 0: This game is the same as the real game. Then, for all $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, the j-th ciphertext and the ℓ-th secret key that $\mathcal{A}$ obtains from the oracles correspond to

$$\{r_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle\right) \in \mathbb{Z}_n, \tag{6.1}$$

$$\tilde{\mathbf{x}}_{i,j} := (\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, 0^m, 1) \in \mathbb{Z}_n^{2m+1}, \quad \tilde{\mathbf{y}}_{i,\ell} := (\mathbf{y}^{\beta}_{i,\ell}, 0^m, r_{i,\ell}) \in \mathbb{Z}_n^{2m+1}.$$

Game 1: This game is the same as Game 0 except that $\tilde{\mathbf{x}}_{i,j}$ in the ciphertext queries is defined as follows:

$$\boxed{\{\mathbf{v}_i\}_{i\in[\mu]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m,} \quad \tilde{\mathbf{x}}_{i,j} := (\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \boxed{\mathbf{v}_i}, 1) \in \mathbb{Z}_n^{2m+1} \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}].$$

Game 2: This game is the same as Game 1 except that $\tilde{\mathbf{y}}_{i,\ell}$ in the secret key queries is defined as follows:

$$\boxed{\{r'_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r'_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]} \left(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle\right)\right) \in \mathbb{Z}_n,} \tag{6.2}$$

$$\tilde{\mathbf{y}}_{i,\ell} := (\mathbf{y}^{\beta}_{i,\ell}, \boxed{\mathbf{y}^{0}_{i,\ell}, r'_{i,\ell}}) \in \mathbb{Z}_n^{2m+1} \quad \text{for all } i \in [\mu] \text{ and } \ell \in [q_{\mathsf{sk}}].$$

Game 3: This game is the same as Game 2 except that $\tilde{\mathbf{x}}_{i,j}$ in the ciphertext queries is defined as follows:

$$\tilde{\mathbf{x}}_{i,j} := (\mathbf{x}^{\beta}_{i,j} \boxed{-\mathbf{x}^{\beta}_{i,1}} + \mathbf{u}_i, \boxed{\mathbf{x}^0_{i,1}} + \mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1} \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}].$$

Game 4: This game is the same as Game 3 except that $\tilde{\mathbf{x}}_{i,j}$ in the ciphertext queries is defined as follows:

$$\tilde{\mathbf{x}}_{i,j} := (\mathbf{u}_i, \boxed{\mathbf{x}^0_{i,j}} + \mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1} \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}].$$

Game 5: This game is the same as Game 4 except that $\tilde{\mathbf{y}}_{i,\ell}$ in the secret key queries is defined as follows:

$$\boxed{\{r''_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r''_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r''_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle\right) \in \mathbb{Z}_n,} \tag{6.3}$$

$$\tilde{\mathbf{y}}_{i,\ell} := (\boxed{0^m}, \mathbf{y}^0_{i,\ell}, \boxed{r''_{i,\ell}}) \in \mathbb{Z}_n^{2m+1} \quad \text{for all } i \in [\mu] \text{ and } \ell \in [q_{\mathsf{sk}}].$$

Thanks to Lemma 6.1 to Lemma 6.6 and the observation in Section 6.2.1, Theorem 6.1 holds. □

In the following, we denote the event that $\mathcal{A}$'s output is equal to β, i.e., $\beta = \beta'$, in Game ι by E_ι.

Lemma 6.1. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_1$ for Priv-IPFE s.t.*

$$|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \leq \mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{B}_1,\mathsf{w\text{-}fh}}(\lambda),$$
$$\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + \mu q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. Let $\delta \in \{0,1\}$ be a random coin that corresponds to β in Definition 3.6, chosen by the game for weakly function-hiding Priv-IPFE. $\mathcal{B}_1$ behaves as follows.

1. $\mathcal{B}_1$ chooses a bit $\beta \xleftarrow{\mathsf{U}} \{0,1\}$ and vectors $\{\mathbf{u}_i\}_{i\in[\mu]}, \{\mathbf{v}_i\}_{i\in[\mu]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m$.
2. $\mathcal{B}_1$ obtains pp' from the game and inputs it to $\mathcal{A}$ as pp.
3. When $\mathcal{A}$ makes a ciphertext query for $(i, (\mathbf{x}^0_{i,j}, \mathbf{x}^1_{i,j}))$, $\mathcal{B}_1$ first sets $\tilde{\mathbf{x}}^0_{i,j} := (\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, 0^m, 1) \in \mathbb{Z}_n^{2m+1}$ and $\tilde{\mathbf{x}}^1_{i,j} := (\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1}$. Then, $\mathcal{B}_1$ queries $\mathcal{O}_{\mathsf{ct}}$ on $(i, (\tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{x}}^1_{i,j}))$ and obtains $\mathsf{ct}'_{i,j}$ from it. Finally, $\mathcal{B}_1$ replies $\mathsf{ct}_{i,j} := \mathsf{ct}'_{i,j}$ to $\mathcal{A}$.
4. When $\mathcal{A}$ makes a secret key query for $(\{\mathbf{y}^0_{i,\ell}\}_{i\in[\mu]}, \{\mathbf{y}^1_{i,\ell}\}_{i\in[\mu]})$, $\mathcal{B}_1$ first sets $\tilde{\mathbf{y}}^0_{i,\ell} = \tilde{\mathbf{y}}^1_{i,\ell} := (\mathbf{y}^{\beta}_{i,\ell}, 0^m, r_{i,\ell}) \in \mathbb{Z}_n^{2m+1}$ where $r_{i,\ell}$ is generated as Eq.(6.1). Then, $\mathcal{B}_1$ queries $\mathcal{O}_{\mathsf{sk}}$ on $(i, (\tilde{\mathbf{y}}^0_{i,\ell}, \tilde{\mathbf{y}}^1_{i,\ell}))$ and obtains $\mathsf{sk}'_{i,\ell}$ from it for all $i \in [\mu]$. Finally, $\mathcal{B}_1$ replies $\mathsf{sk}_\ell := \{\mathsf{sk}'_{i,\ell}\}_{i\in[\mu]}$ to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs β', $\mathcal{B}_1$ outputs the truth value of $(\beta = \beta')$.

In the above description, for all $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^0_{i,\ell}\rangle = \langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell}\rangle = \langle \tilde{\mathbf{x}}^1_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell}\rangle = \langle \mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \mathbf{y}^{\beta}_{i,\ell}\rangle + r_{i,\ell}.$$

Then, $\mathcal{B}_1$ follows the condition Eq.(3.3). It is not difficult to confirm that $\mathcal{A}$'s view corresponds to Game 0 if $\delta = 0$ and Game 1 if $\delta = 1$. This concludes the proof. □

Lemma 6.2. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_2$ for Priv-IPFE s.t.*

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \le \mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{B}_2,\mathsf{w\text{-}fh}}(\lambda),$$
$$\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + \mu q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. Let $\delta \in \{0, 1\}$ be a random coin that corresponds to β in Definition 3.6, chosen by the game for weakly function-hiding Priv-IPFE. $\mathcal{B}_2$ behaves as follows.

1. $\mathcal{B}_2$ chooses a bit $\beta \overset{\mathsf{U}}{\leftarrow} \{0, 1\}$ and vectors $\{\mathbf{u}_i\}_{i\in[\mu]}, \{\mathbf{v}_i\}_{i\in[\mu]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_n^m$.
2. $\mathcal{B}_2$ obtains pp' from the game and inputs it to $\mathcal{A}$ as pp.
3. When $\mathcal{A}$ makes a ciphertext query for $(i, (\mathbf{x}^0_{i,j}, \mathbf{x}^1_{i,j}))$, $\mathcal{B}_2$ first sets $\tilde{\mathbf{x}}^0_{i,j} = \tilde{\mathbf{x}}^1_{i,j} := (\mathbf{x}^\beta_{i,j} + \mathbf{u}_i, \mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1}$. Then, $\mathcal{B}_2$ queries $\mathcal{O}_{\mathsf{ct}}$ on $(i, (\tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{x}}^1_{i,j}))$ and obtains $\mathsf{ct}'_{i,j}$ from it. Finally, $\mathcal{B}_2$ replies $\mathsf{ct}_{i,j} := \mathsf{ct}'_{i,j}$ to $\mathcal{A}$.
4. When $\mathcal{A}$ makes a secret key query for $(\{\mathbf{y}^0_{i,\ell}\}_{i\in[\mu]}, \{\mathbf{y}^1_{i,\ell}\}_{i\in[\mu]})$, $\mathcal{B}_2$ first computes

$$\{r_{i,\ell}\}_{i\in[\mu-1]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_n, \quad r_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle\right) \in \mathbb{Z}_n,$$
$$r'_{i,\ell} := r_{i,\ell} - \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle, \quad \tilde{\mathbf{y}}^0_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, 0^m, r_{i,\ell}) \in \mathbb{Z}_n^{2m+1}, \quad \tilde{\mathbf{y}}^1_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, \mathbf{y}^0_{i,\ell}, r'_{i,\ell}) \in \mathbb{Z}_n^{2m+1}$$
$$\text{for all } i \in [\mu].$$

   Then, $\mathcal{B}_2$ queries $\mathcal{O}_{\mathsf{sk}}$ on $(i, (\tilde{\mathbf{y}}^0_{i,\ell}, \tilde{\mathbf{y}}^1_{i,\ell}))$ and obtains $\mathsf{sk}'_{i,\ell}$ from it for all $i \in [\mu]$. Finally, $\mathcal{B}_2$ replies $\mathsf{sk}_\ell := \{\mathsf{sk}'_{i,\ell}\}_{i\in[\mu]}$ to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs β', $\mathcal{B}_2$ outputs the truth value of $(\beta = \beta')$.

In the above description, for all $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^0_{i,\ell}\rangle = \langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell}\rangle = \langle \tilde{\mathbf{x}}^1_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell}\rangle = \langle \mathbf{x}^\beta_{i,j} + \mathbf{u}_i, \mathbf{y}^\beta_{i,\ell}\rangle + r_{i,\ell}.$$

Then, $\mathcal{B}_2$ follows the condition Eq. (3.3). Observe that $\{r_{i,\ell}\}_{i\in[\mu-1]}$ are chosen randomly from $\mathbb{Z}_n$, then $\{r'_{i,\ell}\}_{i\in[\mu-1]}$ are also random elements in $\mathbb{Z}_n$ from the viewpoint of the adversary. Additionally, we have

$$\begin{aligned} r'_{\mu,\ell} = r_{\mu,\ell} - \langle \mathbf{y}^0_{\mu,\ell}, \mathbf{v}_\mu\rangle &= -\left(\sum_{i\in[\mu-1]} r_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle\right) - \langle \mathbf{y}^0_{\mu,\ell}, \mathbf{v}_\mu\rangle \\ &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]} \left(\langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle\right)\right). \end{aligned}$$

Then, $\mathcal{A}$'s view corresponds to Game 1 if $\delta = 0$ and Game 2 if $\delta = 1$. This concludes the proof. □

Lemma 6.3. *For any PPT adversary $\mathcal{A}$, we have*

$$\Pr[\mathsf{E}_2] = \Pr[\mathsf{E}_3].$$

Proof. Lemma 6.3 follows from Claim 6.1 and Claim 6.2. To prove Lemma 6.3, we use a kind of complexity leveraging argument. In the following, we randomly choose vectors independently from the security game as $\{\hat{\mathbf{x}}^{\gamma}_{i,1}\}_{\gamma\in\{0,1\},i\in[\mu]} \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m$. The purpose is to assure that $\hat{\mathbf{x}}^0_{i,1}$ and $\hat{\mathbf{x}}^1_{i,1}$ are independent from $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{v}}_i$ in Claim 6.2. □

Claim 6.1. *For any PPT adversary $\mathcal{A}$ and both $\iota \in \{2,3\}$, we have*

$$\Pr[\mathsf{E}_\iota] = \Pr[\mathsf{E}_\iota | \{\hat{\mathbf{x}}^{\gamma}_{i,1}\}_{\gamma\in\{0,1\},i\in[\mu]} \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m, \forall\gamma, i,\ \hat{\mathbf{x}}^{\gamma}_{i,1} = \mathbf{x}^{\gamma}_{i,1} \pmod{n}],$$

where $\mathbf{x}^{\gamma}_{i,1} \in \mathbb{Z}^m$ for $\gamma \in \{0,1\}$ and $i \in [\mu]$ is the γ-side vector queried at $\mathcal{A}$'s first ciphertext query for slot i.

Proof. Vectors $\{\hat{\mathbf{x}}^{\gamma}_{i,1}\}_{\gamma\in\{0,1\},i\in[\mu]}$ are chosen independently from $\mathcal{A}$'s view. Then, the event $[\forall\gamma, i,\ \hat{\mathbf{x}}^{\gamma}_{i,1} = \mathbf{x}^{\gamma}_{i,1} \pmod{n}]$ does not affect $\mathcal{A}$'s behavior. □

Claim 6.2. *For any PPT adversary $\mathcal{A}$, we have*

$$\begin{aligned}&\Pr[\mathsf{E}_2 | \{\hat{\mathbf{x}}^{\gamma}_{i,1}\}_{\gamma\in\{0,1\},i\in[\mu]} \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m, \forall\gamma, i,\ \hat{\mathbf{x}}^{\gamma}_{i,1} = \mathbf{x}^{\gamma}_{i,1} \pmod{n}]\\ =&\Pr[\mathsf{E}_3 | \{\hat{\mathbf{x}}^{\gamma}_{i,1}\}_{\gamma\in\{0,1\},i\in[\mu]} \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m, \forall\gamma, i,\ \hat{\mathbf{x}}^{\gamma}_{i,1} = \mathbf{x}^{\gamma}_{i,1} \pmod{n}].\end{aligned}$$

Proof. We redefine $\mathbf{u}_i$ and $\mathbf{v}_i$ as $\mathbf{u}_i := \tilde{\mathbf{u}}_i - \hat{\mathbf{x}}^{\beta}_{i,1}$ and $\mathbf{v}_i := \tilde{\mathbf{v}}_i + \hat{\mathbf{x}}^{0}_{i,1}$ where $\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m$ for all $i \in [\mu]$. Observe that $\mathbf{u}_i$ and $\mathbf{v}_i$ are identically distributed to the original ones, i.e., $\mathbf{u}_i, \mathbf{v}_i \overset{\mathsf{U}}{\longleftarrow} \mathbb{Z}_n^m$. This is because $\hat{\mathbf{x}}^0_{i,1}$ and $\hat{\mathbf{x}}^1_{i,1}$ are chosen independently from $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{v}}_i$. Under the condition such that $\forall\gamma, i,\ \hat{\mathbf{x}}^{\gamma}_{i,1} = \mathbf{x}^{\gamma}_{i,1} \pmod{n}$, we have

(In the secret keys)

$$\begin{aligned} r'_{\mu,\ell} &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle\right)\right)\\ &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^{\beta}_{i,\ell}, \tilde{\mathbf{u}}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \tilde{\mathbf{v}}_i\rangle\right) + \sum_{i\in[\mu]}\left(-\langle \mathbf{y}^{\beta}_{i,\ell}, \hat{\mathbf{x}}^{\beta}_{i,1}\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \hat{\mathbf{x}}^{0}_{i,1}\rangle\right)\right) \qquad (6.4)\\ &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^{\beta}_{i,\ell}, \tilde{\mathbf{u}}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \tilde{\mathbf{v}}_i\rangle\right)\right) \quad \text{for all } \ell \in [q_{\mathsf{sk}}],\end{aligned}$$

(In the ciphertexts)

$$\tilde{\mathbf{x}}_{i,j} = (\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \mathbf{v}_i, 1) = (\mathbf{x}^{\beta}_{i,j} - \hat{\mathbf{x}}^{\beta}_{i,1} + \tilde{\mathbf{u}}_i, \hat{\mathbf{x}}^{0}_{i,1} + \tilde{\mathbf{v}}_i, 1) \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}]. \qquad (6.5)$$

Eq. (6.4) follows from the condition Eq. (3.4) because $\sum_{i\in[\mu]}\left(-\langle \mathbf{y}^{\beta}_{i,\ell}, \hat{\mathbf{x}}^{\beta}_{i,1}\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \hat{\mathbf{x}}^{0}_{i,1}\rangle\right) = 0$. Then, from Eq. (6.4) and Eq. (6.5), $\mathcal{A}$'s views are identical in Game 2 and Game 3 if $\forall\gamma, i,\ \hat{\mathbf{x}}^{\gamma}_{i,1} = \mathbf{x}^{\gamma}_{i,1} \pmod{n}$. This proves Claim 6.2. □

Lemma 6.4. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_3$ for Priv-IPFE s.t.*

$$\begin{aligned}&|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \le \mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{B}_3,\mathsf{w\text{-}fh}}(\lambda),\\ &\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + \mu q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),\end{aligned}$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. We use the following claim in the proof of Lemma 6.4.

Claim 6.3. *For all $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, we have*

$$\langle \mathbf{x}^{\beta}_{i,j} - \mathbf{x}^{\beta}_{i,1}, \mathbf{y}^{\beta}_{i,\ell} \rangle = \langle \mathbf{x}^{0}_{i,j} - \mathbf{x}^{0}_{i,1}, \mathbf{y}^{0}_{i,\ell} \rangle.$$

Proof. From Eq. (3.4), we have

$$\langle \mathbf{x}^{\beta}_{i,j}, \mathbf{y}^{\beta}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{\beta}_{\iota,1}, \mathbf{y}^{\beta}_{\iota,\ell} \rangle = \langle \mathbf{x}^{0}_{i,j}, \mathbf{y}^{0}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{0}_{\iota,1}, \mathbf{y}^{0}_{\iota,\ell} \rangle \tag{6.6}$$

$$\langle \mathbf{x}^{\beta}_{i,1}, \mathbf{y}^{\beta}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{\beta}_{\iota,1}, \mathbf{y}^{\beta}_{\iota,\ell} \rangle = \langle \mathbf{x}^{0}_{i,1}, \mathbf{y}^{0}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{0}_{\iota,1}, \mathbf{y}^{0}_{\iota,\ell} \rangle \tag{6.7}$$

Then Eq. (6.6) − Eq. (6.7) yields Claim 6.3. □

Next, we describe $\mathcal{B}_3$'s behavior. Let $\delta \in \{0,1\}$ be a random coin that corresponds to β in Definition 3.6, chosen by the game for weakly function-hiding Priv-IPFE.

1. $\mathcal{B}_3$ chooses a bit $\beta \xleftarrow{\mathsf{U}} \{0,1\}$ and vectors $\{\mathbf{u}_i\}_{i \in [\mu]}, \{\mathbf{v}_i\}_{i \in [\mu]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m$.
2. $\mathcal{B}_3$ obtains pp' from the game and inputs it to $\mathcal{A}$ as pp.
3. When $\mathcal{A}$ makes a ciphertext query for $(i, (\mathbf{x}^0_{i,j}, \mathbf{x}^1_{i,j}))$, $\mathcal{B}_3$ first sets $\tilde{\mathbf{x}}^0_{i,j} := (\mathbf{x}^{\beta}_{i,j} - \mathbf{x}^{\beta}_{i,1} + \mathbf{u}_i, \mathbf{x}^0_{i,1} + \mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1}$ and $\tilde{\mathbf{x}}^1_{i,j} := (\mathbf{u}_i, \mathbf{x}^0_{i,j} + \mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1}$. Then, $\mathcal{B}_3$ queries O_{ct} on $(i, (\tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{x}}^1_{i,j}))$ and obtains $\mathsf{ct}'_{i,j}$ from it. Finally, $\mathcal{B}_3$ replies $\mathsf{ct}_{i,j} := \mathsf{ct}'_{i,j}$ to $\mathcal{A}$.
4. When $\mathcal{A}$ makes a secret key query for $(\{\mathbf{y}^0_{i,\ell}\}_{i \in [\mu]}, \{\mathbf{y}^1_{i,\ell}\}_{i \in [\mu]})$, $\mathcal{B}_3$ first sets $\tilde{\mathbf{y}}^0_{i,\ell} = \tilde{\mathbf{y}}^1_{i,\ell} := (\mathbf{y}^{\beta}_{i,\ell}, \mathbf{y}^0_{i,\ell}, r'_{i,\ell}) \in \mathbb{Z}_n^{2m+1}$ where $r'_{i,\ell}$ is generated as Eq. (6.2). Then, $\mathcal{B}_3$ queries O_{sk} on $(i, (\tilde{\mathbf{y}}^0_{i,\ell}, \tilde{\mathbf{y}}^1_{i,\ell}))$ and obtains $\mathsf{sk}'_{i,\ell}$ from it for all $i \in [\mu]$. Finally, $\mathcal{B}_3$ replies $\mathsf{sk}_\ell := \{\mathsf{sk}'_{i,\ell}\}_{i \in [\mu]}$ to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs β', $\mathcal{B}_3$ outputs the truth value of $(\beta = \beta')$.

In the above description, for all $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\begin{aligned}
\langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^0_{i,\ell} \rangle &= \langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell} \rangle \\
&= \langle \mathbf{x}^{\beta}_{i,j} - \mathbf{x}^{\beta}_{i,1}, \mathbf{y}^{\beta}_{i,\ell} \rangle + \langle \mathbf{u}_i, \mathbf{y}^{\beta}_{i,\ell} \rangle + \langle \mathbf{x}^0_{i,1} + \mathbf{v}_i, \mathbf{y}^0_{i,\ell} \rangle + r'_{i,\ell} \\
&= \langle \mathbf{u}_i, \mathbf{y}^{\beta}_{i,\ell} \rangle + \langle \mathbf{x}^0_{i,j} - \mathbf{x}^0_{i,1}, \mathbf{y}^0_{i,\ell} \rangle + \langle \mathbf{x}^0_{i,1} + \mathbf{v}_i, \mathbf{y}^0_{i,\ell} \rangle + r'_{i,\ell} \\
&= \langle \mathbf{u}_i, \mathbf{y}^{\beta}_{i,\ell} \rangle + \langle \mathbf{x}^0_{i,j} + \mathbf{v}_i, \mathbf{y}^0_{i,\ell} \rangle + r'_{i,\ell} \\
&= \langle \tilde{\mathbf{x}}^1_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell} \rangle.
\end{aligned}$$

In the third line, we use Claim 6.3. Then, $\mathcal{B}_3$ follows the condition Eq. (3.3). It is not difficult to confirm that $\mathcal{A}$'s view corresponds to Game 3 if $\delta = 0$ and Game 4 if $\delta = 1$. This concludes the proof. □

Lemma 6.5. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_4$ for Priv-IPFE s.t.*

$$\begin{aligned}
&|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_5]| \leq \mathsf{Adv}^{\mathsf{Priv\text{-}IPFE}}_{\mathcal{B}_4,\mathsf{w\text{-}fh}}(\lambda), \\
&\mathsf{Time}(\mathcal{B}_4) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + \mu q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),
\end{aligned}$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

Proof. Let $\delta \in \{0,1\}$ be a random coin that corresponds to β in Definition 3.6, chosen by the game for weakly function-hiding Priv-IPFE. $\mathcal{B}_4$ behaves as follows.

1. $\mathcal{B}_4$ chooses a bit $\beta \overset{\mathsf{U}}{\leftarrow} \{0,1\}$ and vectors $\{\mathbf{u}_i\}_{i\in[\mu]}, \{\mathbf{v}_i\}_{i\in[\mu]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_n^m$.
2. $\mathcal{B}_4$ obtains pp' from the game and inputs it to $\mathcal{A}$ as pp.
3. When $\mathcal{A}$ makes a ciphertext query for $(i, (\mathbf{x}^0_{i,j}, \mathbf{x}^1_{i,j}))$, $\mathcal{B}_4$ first sets $\tilde{\mathbf{x}}^0_{i,j} = \tilde{\mathbf{x}}^1_{i,j} := (\mathbf{u}_i, \mathbf{x}^0_{i,j}+\mathbf{v}_i, 1) \in \mathbb{Z}_n^{2m+1}$. Then, $\mathcal{B}_4$ queries O_{ct} on $(i, (\tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{x}}^1_{i,j}))$ and obtains $\mathsf{ct}'_{i,j}$ from it. Finally, $\mathcal{B}_4$ replies $\mathsf{ct}_{i,j} := \mathsf{ct}'_{i,j}$ to $\mathcal{A}$.
4. When $\mathcal{A}$ makes a secret key query for $(\{\mathbf{y}^0_{i,\ell}\}_{i\in[\mu]}, \{\mathbf{y}^1_{i,\ell}\}_{i\in[\mu]})$, $\mathcal{B}_4$ first computes
$$\{r'_{i,\ell}\}_{i\in[\mu-1]} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_n,\ \ r'_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle\right)\right) \in \mathbb{Z}_n,$$
$$r''_{i,\ell} := r'_{i,\ell} + \langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle,\ \ \tilde{\mathbf{y}}^0_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, \mathbf{y}^0_{i,\ell}, r'_{i,\ell}) \in \mathbb{Z}_n^{2m+1},\ \ \tilde{\mathbf{y}}^1_{i,\ell} := (0^m, \mathbf{y}^0_{i,\ell}, r''_{i,\ell}) \in \mathbb{Z}_n^{2m+1}$$
for all $i \in [\mu]$.

   Then, $\mathcal{B}_4$ queries O_{sk} on $(i, (\tilde{\mathbf{y}}^0_{i,\ell}, \tilde{\mathbf{y}}^1_{i,\ell}))$ and obtains $\mathsf{sk}'_{i,\ell}$ from it for all $i \in [\mu]$. Finally, $\mathcal{B}_4$ replies $\mathsf{sk}_\ell := \{\mathsf{sk}'_{i,\ell}\}_{i\in[\mu]}$ to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs β', $\mathcal{B}_4$ outputs the truth value of $(\beta = \beta')$.

In the above description, for all $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, we have
$$\langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^0_{i,\ell}\rangle = \langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell}\rangle = \langle \tilde{\mathbf{x}}^1_{i,j}, \tilde{\mathbf{y}}^1_{i,\ell}\rangle = \langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{x}^0_{i,j} + \mathbf{v}_i, \mathbf{y}^0_{i,\ell}\rangle + r'_{i,\ell}.$$
Then, $\mathcal{B}_4$ follows the condition Eq. (3.3). Observe that $\{r'_{i,\ell}\}_{i\in[\mu-1]}$ are chosen randomly from $\mathbb{Z}_n$, then $\{r''_{i,\ell}\}_{i\in[\mu-1]}$ are also random elements in $\mathbb{Z}_n$ from the viewpoint of the adversary. Additionally, we have
$$\begin{aligned} r''_{\mu,\ell} = r'_{\mu,\ell} + \langle \mathbf{y}^\beta_{\mu,\ell}, \mathbf{u}_\mu\rangle &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle\right)\right) + \langle \mathbf{y}^\beta_{\mu,\ell}, \mathbf{u}_\mu\rangle \\ &= -\left(\sum_{i\in[\mu-1]} r''_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle\right). \end{aligned}$$
Then, $\mathcal{A}$'s view corresponds to Game 4 if $\delta = 0$ and Game 5 if $\delta = 1$. This concludes the proof. □

Lemma 6.6. *For any PPT adversary $\mathcal{A}$, we have*
$$\Pr[\mathsf{E}_5] = 1/2.$$
Lemma 6.6 is trivial because $\mathcal{A}$ does not obtain any information about β in Game 5.

6.2.1 Optimization

On Lemma 6.1 and Lemma 6.2, we define that

$$\tilde{\mathbf{x}}^0_{i,j} := (\mathbf{x}^\beta_{i,j} + \mathbf{u}_i, 0^m, 1), \quad \tilde{\mathbf{x}}^1_{i,j} := (\mathbf{x}^\beta_{i,j} + \mathbf{u}_i, \mathbf{v}_i, 1),$$
$$\tilde{\mathbf{y}}^0_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, 0^m, r_{i,\ell}), \quad \tilde{\mathbf{y}}^1_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, \mathbf{y}^0_{i,\ell}, r'_{i,\ell}).$$

Then, we have $\langle \tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{y}}^0_{i,j} \rangle = \langle \tilde{\mathbf{x}}^1_{i,j}, \tilde{\mathbf{y}}^0_{i,j} \rangle = \langle \tilde{\mathbf{x}}^1_{i,j}, \tilde{\mathbf{y}}^1_{i,j} \rangle$ for all $i \in [\mu]$, $j \in [q_{\mathrm{ct},i}]$, and $\ell \in [q_{\mathrm{sk}}]$, which satisfies the condition Eq.(3.3). Hence, we do not need Game 1 actually and can prove that

$$|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_2]| \le \mathsf{Adv}^{\text{Priv-IPFE}}_{\mathcal{B}_1,\text{w-fh}}(\lambda).$$

Similarly, we can also prove that

$$|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_5]| \le \mathsf{Adv}^{\text{Priv-IPFE}}_{\mathcal{B}_2,\text{w-fh}}(\lambda).$$

6.3 Application to Our Scheme

Applying the conversion to our scheme presented in Section 5.1, we can obtain a tightly secure fully function-hiding MIPFE scheme. First, we confirm that our scheme satisfies the property presented in Section 6.1.

1. Theorem 5.1 says that our scheme is weakly function-hiding.
2. We can define that $n := p$, $G := G_T$, and $E : a \in \mathbb{Z}_p \longrightarrow [a]_T \in G_T$. The group law $\circ$ corresponds to the multiplication over G_T.
3. We can define that Dec_1 computes $[d]_T$ and Dec_2 searches for the discrete logarithm of $[d]_T$.
4. It is obvious that $g_T^a \cdot g_T^b = g_T^{a+b}$.

Then, from Theorem 5.1 and Theorem 6.1, we obtain the following corollary.

Corollary 6.1. *Let MIPFE be the MIPFE scheme obtained by applying the conversion in Section 6.1 to our weakly function-hiding Priv-IPFE scheme. Then MIPFE is fully function-hiding. More formally, let μ be a number of slots, $q_{\mathrm{ct}} := \sum_{i\in[\mu]} q_{\mathrm{ct},i}$ be the total number of the ciphertext queries by $\mathcal{A}$, q_{sk} be the total number of the secret key queries by $\mathcal{A}$, and m be a vector length. Then, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exist PPT adversaries $\mathcal{B}_1, \ldots, \mathcal{B}_4$ for the $\mathcal{D}_k$-MDDH and we have*

$$\mathsf{Adv}^{\mathsf{MIPFE}}_{\mathcal{A},\mathsf{f\text{-}fh}}(\lambda) \le 8 \sum_{\iota\in\{1,2\}} \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B}_\iota,\mathsf{BG},1}(\lambda) + 8 \sum_{\iota\in\{3,4\}} \mathsf{Adv}^{\mathcal{D}_k\text{-MDDH}}_{\mathcal{B}_\iota,\mathsf{BG},2}(\lambda) + 2^{-\Omega(\lambda)},$$
$$\max_{\iota\in[4]}\{\mathsf{Time}(\mathcal{B}_\iota)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathrm{ct}} + \mu q_{\mathrm{sk}})\mathsf{poly}(\lambda, m),$$

where $\mathsf{poly}(\lambda, m)$ *is independent from* $\mathsf{Time}(\mathcal{A})$.

References

[1] Genbank and wgs statistics. https://www.ncbi.nlm.nih.gov/genbank/statistics/.

[2] 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.

[3] M. Abdalla, D. Catalano, D. Fiore, R. Gay, and B. Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 597–627. Springer, Heidelberg, Aug. 2018.

[4] M. Abdalla, R. Gay, M. Raykova, and H. Wee. Multi-input inner-product functional encryption from pairings. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 601–626. Springer, Heidelberg, Apr. / May 2017.

[5] M. Abe, D. Hofheinz, R. Nishimaki, M. Ohkubo, and J. Pan. Compact structure-preserving signatures with almost tight security. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 548–580. Springer, Heidelberg, Aug. 2017.

[6] 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.

[7] N. Attrapadung, G. Hanaoka, and S. Yamada. A framework for identity-based encryption with almost tight security. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 521–549. Springer, Heidelberg, Nov. / Dec. 2015.

[8] 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.

[9] C. E. Z. Baltico, D. Catalano, D. Fiore, and R. Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 67–98. Springer, Heidelberg, Aug. 2017.

[10] P. S. L. M. Barreto, C. Costello, R. Misoczki, M. Naehrig, G. C. C. F. Pereira, and G. Zanon. Subgroup security in pairing-based cryptography. In K. E. Lauter and F. Rodríguez-Henríquez, editors, *LATINCRYPT 2015*, volume 9230 of *LNCS*, pages 245–265. Springer, Heidelberg, Aug. 2015.

[11] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

[12] 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.

[13] 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.

[14] 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.

[15] Z. Brakerski and G. Segev. Function-private functional encryption in the private-key setting. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 306–324. Springer, Heidelberg, Mar. 2015.

[16] G. Castagnos, F. Laguillaumie, and I. Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In T. Peyrin and S. Galbraith, editors, *ASIACRYPT 2018, Part II*, LNCS, pages 733–764. Springer, Heidelberg, Dec. 2018.

[17] J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Heidelberg, Aug. 2013.

[18] 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.

[19] P. Datta, T. Okamoto, and J. Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption. In M. Abdalla and R. Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 245–277. Springer, Heidelberg, Mar. 2018.

[20] A. Enge and J. Milan. Implementing cryptographic pairings at standard security levels. In R. S. Chakraborty, V. Matyas, and P. Schaumont, editors, *Security, Privacy, and Applied Cryptography Engineering - 4th International Conference, SPACE 2014, Pune, India, October 18-22, 2014. Proceedings*, volume 8804 of *Lecture Notes in Computer Science*, pages 28–46. Springer, 2014.

[21] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, Jan. 2017.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] R. Gay, D. Hofheinz, and L. Kohl. Kurosawa-desmedt meets tight security. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 133–160. Springer, Heidelberg, Aug. 2017.

[27] R. Gay, L. Kowalczyk, and H. Wee. Tight adaptively secure broadcast encryption with short ciphertexts and keys. In D. Catalano and R. De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 123–139. Springer, Heidelberg, Sept. 2018.

[28] 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.

[29] D. Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 251–281. Springer, Heidelberg, Jan. 2016.

[30] D. Hofheinz. Adaptive partitioning. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 489–518. Springer, Heidelberg, Apr. / May 2017.

[31] D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, Aug. 2012.

[32] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu. Function-hiding inner product encryption is practical. In D. Catalano and R. De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 544–562. Springer, Heidelberg, Sept. 2018.

[33] B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, Nov. / Dec. 2015.

[34] H. Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 599–629. Springer, Heidelberg, Aug. 2017.

[35] H. Lin and V. Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In I. Dinur, editor, *57th FOCS*, pages 11–20. IEEE Computer Society Press, Oct. 2016.

[36] M. Naor and O. Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

[37] A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. `http://eprint.iacr.org/2010/556`.

[38] J. Tomida, M. Abe, and T. Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In M. Bishop and A. C. A. Nascimento, editors, *ISC 2016*, volume 9866 of *LNCS*, pages 408–425. Springer, Heidelberg, Sept. 2016.

[39] B. Waters. A punctured programming approach to adaptively secure functional encryption. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 678–697. Springer, Heidelberg, Aug. 2015.

A Function-Hiding IPFE Scheme from AGRW17 and Lin17

Lin presented a paradigm to construct a weakly function-hiding IPFE scheme from an IPFE scheme and a concrete instantiation [34]. However, the presented scheme is selectively secure because the underlying one is. We can easily obtain an adaptively secure one by applying her paradigm to the adaptively secure IPFE scheme in [4]. The resulting scheme is as follows.

Setup$(1^\lambda, 1^m)$**:** It takes a security parameter 1^λ and a vector length 1^m. Then, it outputs a public parameter pp and a master secret key msk as follows.

$$\mathbb{G}_{\mathsf{BG}} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathsf{pp} := \mathbb{G}_{\mathsf{BG}},$$
$$\mathbf{A}, \mathbf{D} \leftarrow \mathcal{D}_k, \quad \mathbf{W} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{m \times (k+1)}, \quad \mathbf{V} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^{(m+k+1)\times(k+1)}, \quad \mathsf{msk} := (\mathbf{A}, \mathbf{D}, \mathbf{W}, \mathbf{V}).$$

Enc(pp, msk, $\mathbf{x}$)**:** It takes pp, msk, and $\mathbf{x} \in \mathbb{Z}^m$ and outputs a ciphertext ct as follows.

$$\mathbf{s} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k, \quad \mathbf{c}_{\mathsf{in}} := (\mathbf{A}\mathbf{s}, \mathbf{W}\mathbf{A}\mathbf{s} + \mathbf{x}) \in \mathbb{Z}_p^{m+k+1},$$
$$\mathbf{k}_{\mathsf{out},1} := -\mathbf{V}^\top \mathbf{c}_{\mathsf{in}} \in \mathbb{Z}_p^{k+1}, \quad \mathbf{k}_{\mathsf{out},2} := \mathbf{c}_{\mathsf{in}}, \quad \mathsf{ct} := ([\mathbf{k}_{\mathsf{out},1}]_1, [\mathbf{k}_{\mathsf{out},2}]_1).$$

KeyGen(pp, msk, $\mathbf{y}$)**:** It takes pp, msk, and $\mathbf{y} \in \mathbb{Z}^m$ and outputs a secret key sk as follows.

$$\mathbf{k}_{\mathsf{in}} := (-\mathbf{W}^\top \mathbf{y}, \mathbf{y}) \in \mathbb{Z}_p^{m+k+1}, \quad \mathbf{r} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^k,$$
$$\mathbf{c}_{\mathsf{out},1} := \mathbf{D}\mathbf{r} \in \mathbb{Z}_p^{k+1}, \quad \mathbf{c}_{\mathsf{out},2} := \mathbf{V}\mathbf{D}\mathbf{r} + \mathbf{k}_{\mathsf{in}} \in \mathbb{Z}_p^{m+k+1}, \quad \mathsf{sk} := ([\mathbf{c}_{\mathsf{out},1}]_2, [\mathbf{c}_{\mathsf{out},2}]_2).$$

Dec(pp, ct, sk)**:** It takes pp, ct, and sk. Then it computes $[d]_T := e([\mathbf{k}_{\mathsf{out},1}]_1, [\mathbf{c}_{\mathsf{out},1}]_2) e([\mathbf{k}_{\mathsf{out},2}]_1, [\mathbf{c}_{\mathsf{out},2}]_2)$ and searches for d exhaustively in the range of $-mX_\lambda Y_\lambda$ to $mX_\lambda Y_\lambda$. If such d is found, it outputs d. Otherwise, it outputs $\perp$.