Towards Practical Inner Product Functional Encryption

Junichi Tomida

Abstract

Functional encryption (FE) is an advanced cryptographic paradigm where we can compute function values from encrypted data without revealing any additional information on the original data. Inner product functional encryption (IPFE) is a class of FE that supports inner products as computable functions. IPFE can be efficiently constructed from well-studied cryptographic tools and thus is practical, unlike FE for general functions. In this thesis, we improve three essential factors in practical applications of IPFE, namely, efficiency, functionality, and security.

First, we present an IPFE scheme with the function-hiding property that is more efficient than previous schemes. The function-hiding property allows us to hide functions that are applied to encrypted data from evaluators and is necessary for some applications. Thus, efficiency improvement is also important for such applications.

Second, we study unboundedness of IPFE. Prior to our work, all IPFE schemes are bounded, that is, the data size used in the schemes has to be fixed. However, cryptosystems normally handle data with various sizes, and thus the bounded property is inconvenient. We solved this problem by introducing unbounded IPFE, in which we can use data with various sizes for encryption and function evaluations.

Finally, we study tight security of IPFE. In the real world, it is natural that malicious users obtain many ciphertexts to break cryptosysetems. Tight security theoretically guarantees that the security of the cryptosystem is independent of the number of ciphertexts that the malicious user obtains. Thus, tight security is practically important. We present the first tightly secure IPFE schemes and show that our schemes are compatible with several conversions for IPFE. Via these conversions, we can also obtain tightly secure IPFE schemes with various extended properties.

Contents

Chapter 1

Background

1.1 Modern Cryptography

Nowadays, companies provide various services on the Internet such as banking, shopping, and video streaming. Cryptography plays an essential role to protect these services from malicious users. For instance, when we register our personal information or credit card numbers to service providers, we encrypt them before sending it to prevent the malicious users from obtaining the information. When a company provides a service to users, the company authenticates users to provide the service to only users who have paid for it. Various cryptographic tools such as encryption, digital signatures, and hash functions are used to securely implement these communications on the Internet.

Public-Key Encryption. One of such central tools in modern cryptography is public-key encryption (PKE). The idea of PKE was first introduced by Diffie and Hellman [DH76] and the first instantiation of PKE was proposed by Rivest, Shamir, and Adleman [RSA78]. Public-key encryption enables a sender to securely send a message to a receiver via an insecure communication channel, where the message can be eavesdropped on, without any preparation between them. In PKE, the receiver generates two keys (a public key and a secret key) and makes the former public. The sender can encrypt a message with the public key, and only the owner of the corresponding secret key can decrypt the encrypted message. Eavesdroppers cannot read the encrypted message since they do not have the secret key.

How to Define Security. One of the most important issues in cryptography is how to define the security requirement of cryptosystems. In modern cryptography, we often use game-based security definitions. The security of many cryptographic primitives such as PKE, digital signatures, hash functions, and pseudorandom generators are defined by game-based definitions. In a game-based definition, we consider a game where an adversary modeled as a probabilistic polynomial-time (PPT) Turing machine tries to break a cryptosystem. Basically, the cryptosystem is said to be secure if no PPT Turing machines can win the game with meaningful probability.

Let us give an example of PKE. One standard security requirement for PKE is indistinguishability under chosen-plaintext attacks (IND-CPA) [GM82], which is defined by the following game. The adversary is first given a public key and chooses two messages m_0 and m_1. Then, the adversary is given a challenge ciphertext of m_β, where $\beta \in \{0, 1\}$ is chosen uniformly at random. Finally, the adversary guesses β and wins if the guess is correct. Roughly speaking, the PKE scheme is said to be IND-CPA secure if, for all PPT adversaries, the probability that the adversary wins is very close to 1/2.

How to Prove Security. Once the security definition is settled, the remaining thing is to prove that the PKE scheme satisfies the definition. However, inventing an unconditionally IND-CPA secure PKE scheme proves $\mathsf{P} \neq \mathsf{NP}$ [Imp95], which is a fundamental open problem in complexity theory. To circumvent this difficulty, we typically prove the security of a cryptosystem under assumptions that some mathematical problems are hard to solve. More precisely, we prove the existence of a polynomial-time Turing reduction from the hard problem to the problem of breaking the cryptosystem. If the hardness of the underlying problem is solid, the proof gives a strong guarantee that breaking the cryptosystem is hard. A notable example is the ElGamal encryption [ElG85], which is proved to be IND-CPA secure under the assumption that the decisional Diffie-Hellman (DDH) problem is hard.

Standard Assumptions. Taking the above proof strategy, reliable assumptions must be used, since the proof is meaningless if the assumption is false. We consider that assumptions that are well-studied and have experienced the test of time are reliable and "standard". It is far more desirable to prove the security of cryptosytems under standard assumptions rather than under ad-hoc assumptions. There are no rigid definitions for standard assumptions, while cryptographers roughly agree about which assumptions are standard. For instance, the assumptions that the factoring, DDH problem [Bon98], and learning with error (LWE) problem [Reg05] are not solvable in polynomial time are considered as standard.

The matrix decisional Diffie-Hellman (MDDH) assumption [EHK+17], which we use entirely in this thesis, is also one of standard assumptions. The MDDH assumption is a family of assumptions and subsumes many specific assumptions. The well-studied standard DDH assumption [Bon98] is the weakest assumption in the MDDH family and thus all assumptions in the MDDH family are considered as standard assumptions. The k-Lin and symmetric external Diffie-Hellman (SXDH) assumptions, which are also used in this thesis, are captured as special cases of the MDDH assumption.

We also briefly mention quantum-safe assumptions. Several number theoretic assumptions, such as factoring and DDH, are broken by quantum algorithms [Sho97]. Although we do not know when such practical quantum computers that can break the number theoretic assumptions with practical parameters will be developed, we have been extensively studying cryptography based on assumptions that are considered to be solid even under the existence of practical quantum computers. The LWE assumption is considered to be one of the most hopeful quantum-safe assumptions, which is based on a well-studied problem in lattices.

1.2 Functional Encryption

Since the first introduction of PKE and digital signatures [DH76, RSA78], a bunch of more advanced and generalized cryptographic paradigms have been proposed to solve problems that are not solvable with such traditional cryptographic tools. One of the main problems in traditional encryption paradigms is that we cannot utilize encrypted data without decrypting it since the encrypted data do not leak any information on the original data. Functional encryption (FE) is an advanced cryptographic paradigm that is expected to solve the unavailability of encrypted data in traditional encryption paradigms.

Although FE was formally defined in the early 2010s [O'N10, BSW11], PKE has been gradually generalized until the notion of FE was introduced. We briefly review the history of FE.

Identity-Based Encryption. The first generalized notion of PKE toward FE is identity-based encryption (IBE), the model of which was introduced by Shamir [Sha84] and achieved by Boneh and Franklin [BF01] and Cocks [Coc01]. Identity-based encryption enables us to attach an identity to a ciphertext and a secret key. Basically, there are three types of keys in IBE (and more generalized notions such as attribute-based encryption and functional encryption, which we explain later), namely, public key, master secret key, and secret key. A public key is used to encrypt messages together with identities, a master secret key is used to generate secret keys that are associated with identities, and a secret key is used to decrypt ciphertexts. A ciphertext for identity i is decryptable with a secret key for identity i' if and only if $i = i'$. A public-key is assumed to be accessible to all users while a master secret key is assumed to belong to an authority who controls accessibility to encrypted data by distributing appropriate secret keys to users.

The advantage of IBE is that we can use identities as public keys, which can simplify public-key management. In IBE system, the user with identity i is given a secret key for identity i from an authority that has a master secret key. A user who wants to send a message to user i encrypts the message with identity i. Then, the ciphertext is decryptable by only a user who has a secret key for i, which is user i.

Attribute-Based Encryption. Attribute-based encryption (ABE), a more generalized notion of PKE, was introduced by Sahai and Waters [SW05, GPSW06]. Each ABE scheme has a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are a ciphertext attribute space and secret-key attribute space, respectively. In ABE, a ciphertext and a secret key can be associated with any ciphertext attribute $x \in \mathcal{X}$ and any secret-key attribute $y \in \mathcal{Y}$, respectively, instead of identities. The ciphertext for x is decryptable with the secret key for y if and only if $\mathsf{P}(x, y) = 1$. For instance, let $\mathcal{X}$ be a set of n-bit binary strings, $\mathcal{Y}$ be a set of n-input Boolean formulae, and $\mathsf{P}(x, f) = f(x)$ where $x \in \mathcal{X}$ and $f \in \mathcal{Y}$. Then, an ABE scheme for P allows us to control access with Boolean formulae over encrypted data [GPSW06].

In addition to Boolean formulae, ABE schemes have been proposed for various computation models, such as span programs [OSW07], (non-)deterministic finite automata [Wat12, AMY19], polynomial-size circuits [GVW13], polynomial-time Turing machines [GKP+13]. Note that IBE can be seen as a subclass of ABE by setting $\mathcal{X}, \mathcal{Y}$ as some identity space, and $\mathsf{P}(x, y) = 1 \Leftrightarrow x = y$.

Predicate encryption (PE) is a similar paradigm to ABE, although it additionally requires that a ciphertext hides the attached attribute as well as the underlying message [KSW08]. There are two levels for the attribute-hiding property, namely, weakly attribute-hiding and fully attribute-hiding. The weakly attribute-hiding property assures that the ciphertext hides its attribute only when the decryptor does not have a key that can decrypt the ciphertext. Weakly attribute-hiding PE schemes are constructed for orthogonality of inner products and general circuits from standard assumptions [OT09, OT10, GVW15].

On the other hand, the fully attribute-hiding property assures that the ciphertext hides its attribute in any condition. As explained in the next paragraph, fully attribute-hiding PE for predicate P is equivalent to FE for functions $f(\cdot) = \mathsf{P}(\cdot, y)$ where $y \in \mathcal{Y}$, and it is hard to construct from standard assumptions. The only fully attribute-hiding PE achieved from standard assumptions is for orthogonality of inner products [KSW08, OT12a, CGW18].

Functional Encryption. Functional encryption, formally defined in [O'N10, BSW11], is a further generalized notion of PKE, which subsumes ABE and PE as a subclass. In an FE scheme that supports function class $\mathcal{F}$, a public key is used to encrypt messages, a master secret key is used to generate secret keys that are associated with functions in $\mathcal{F}$, and a secret key for f is used to compute a function f on encrypted messages by decryption, that is, decryption of a ciphertext for message m with a secret key associated with a function f reveals $f(m)$ and nothing else.

Observe that we can capture ABE and PE as a subclass of FE by considering that $\mathcal{M}^{\mathsf{FE}} = \mathcal{M}^{\mathsf{ABE}} \times \mathcal{X}$, where $\mathcal{M}^{\mathsf{FE}}$ and $\mathcal{M}^{\mathsf{ABE}}$ are message spaces for the FE and ABE schemes, respectively, and $\mathcal{F}$ consists of functions $f : \mathcal{M}^{\mathsf{ABE}} \times \mathcal{X} \to \mathcal{M}^{\mathsf{ABE}} \cup \{\perp\}$ such that

$$f(m, x) = \begin{cases} m & (\mathsf{P}(x, y) = 1) \\ \perp & (\mathsf{P}(x, y) = 0) \end{cases}.$$

We can see the equivalence between fully attribute-hiding PE and FE from the above observation. FE implies fully attribute-hiding PE since attribute x is a part of a message, and decryption never reveals x. For the opposite direction, a decryptor can learn $P(x, y)$ by interpreting correct decryption as $P(x, y) = 1$ and otherwise $P(x, y) = 0$.

Security Definitions for Functional Encryption. Before reviewing the related works on FE, we briefly recall security notions on FE since the familiarity to these notions are helpful to follow the related works. A primal security requirement for FE is the message-hiding property, that is, ciphertexts need to hide the underlying messages. Other than that, an important security property required for FE is collusion resistance. Roughly speaking, it requires that an owner of multiple secret keys cannot compose a new secret key that is "stronger" than the collection of the original secret keys. That is, no PPT adversaries that possess ciphertext for an unknown message m and secret keys for $f_1, \ldots, f_n$ can learn information on m more than $\{f_i(m)\}_{1 \le i \le n}$.

This security requirement is defined basically in two ways: the indistinguishability-based definition (similarly to the case of PKE in Section 1.1) and simulation-based definition. Roughly speaking, the simulation-based definition says that a real ciphertext can be simulated without

knowing the original message, and the simulated ciphertext is indistinguishable from the real ciphertext. More precisely, the definition requires the existence of a PPT algorithm (simulator) that outputs a simulated ciphertext without taking a message m, where the probability distribution of the simulated ciphertext is indistinguishable from that of the real ciphertext for m. Since the simulated ciphertext does not contain the information on m, the simulation-based definition implies that the real ciphertext, which is indistinguishable from the simulated ciphertext, does not leak any information on m to PPT adversaries. On the other hand, the indistinguishability-based definition requires that no PPT adversaries can distinguish two ciphertexts of known messages such as IND-CPA.

Although the simulation-based definition is basically stronger than the indistinguishability-based definition, it is hard to achieve and impossible for some function classes [BSW11, AGVW13]. Throughout this thesis, we use the indistinguishability-based definition. The security game for the indistinguishability-based security of FE is defined almost the same as the IND-CPA security for PKE except that the adversary can query a secret-key oracle on f such that $f(m_0) = f(m_1)$ before and after obtaining the challenge ciphertext. The secret key oracle takes $f \in \mathcal{F}$ and returns a secret key for f. An FE scheme is said to be adaptively or fully secure if the definition is satisfied in the above game. Intuitively, this security definition indicates that an adversary that has the ciphertext for m_b and secret keys for $f_1, \ldots, f_n$ can learn $f_1(m_b), \ldots, f_n(m_b)$ about m_b and nothing else as otherwise the adversary can distinguish the ciphertext for m_0 from that for m_1. It means that the ciphertext does not leak any information on m_b other than decryption values.

When we find that proving adaptive security is too hard, we sometimes use weaker security definitions: selective or semi-adaptive security. Roughly speaking, the adversary is allowed to make a secret-key query only after obtaining the challenge ciphertexts in these security definitions. However, these definitions are somewhat artificial and less desirable than adaptive security. These security notions (i.e., selective, semi-adaptive, and adaptive) are similarly defined for simulation-based security.

Another important security property for FE is the function-hiding property, which requires that secret keys hide their associated functions as ciphertexts hides the messages. The function-hiding property is important in some applications of FE, as explained in Section 2.1. Generally, the function-hiding property is difficult to achieve in the public-key setting, where all users can encrypt any messages with a public key [BRS13a, BRS13b]. This is because an adversary can encrypt any messages and examine what kind of function their secret key has by decrypting the ciphertexts. In other words, an owner of a secret key for some unknown function f can compute $f(m)$ for arbitrary m, and it leaks information of f. In contrast, the function-hiding property can basically be achieved in the private-key setting, where encryption needs some secret information [AAB+15, BS15].

Related Works on Functional Encryption. Functional encryption can be basically classified into two categories with respect to its function class.

General functionalities: This category consists of FE schemes for general circuits or Turing machines. Although they are powerful enough to handle all functions computable in polynomial time, known schemes are built on quite heavy cryptographic primitives and not practical at

all.

Specific functionalities: The second category covers FE schemes for specific functions such as inner products and quadratic functions. They are aimed at obtaining more practical features i.e., efficiency and concrete security, while sacrificing the generality. Therefore, most of them have simple constructions, and their security is based on standard assumptions.

We give the background on the first category in the rest of this section and defer presenting the background on the second category to Section 2.1.

Early works of FE mostly studied the feasibility of simulation-based security for FE. Beneh *et al.* showed that simulation-based security is unachievable even for a family of ABE in the adaptive and many-ciphertexts setting [BSW11]. The many-ciphertext setting is a model where an adversary can obtain many ciphertexts. Agrawal *et al.* gave another result for impossibility on simulation-based security, that is, FE for a function family implying weak pseudorandom functions basically cannot achieve simulation-based security [AGVW13].

The first breakthrough in FE was brought by Garg *et al.* [GGH+13], who proposed the first candidate of FE for all circuits. This is a quite strong result since the function class for circuits is the most general class of computation, which enables us to handle all functions that are computable in polynomial time. In their work, they constructed the first candidate of indistinguishability obfuscation (iO) for all circuits and then show that FE for all circuits is constructible from iO.

Indistinguishable obfuscation is also a very interesting and important cryptographic object. The notion of iO has been known since 2001 [BGI+01], but the first candidate was not proposed more than a decade. Roughly speaking, iO is an algorithm for obfuscating a program, or in other words, converting a program into a black box. That is, an obfuscated program has the same input-output behavior as the original program while the "code" of the original program is not revealed from the obfuscated program.

Since the security of the FE scheme by Garg *et al.* is proven in the selective-security model, cryptographers made a significant effort to construct an adaptively secure FE scheme for general circuits. Boyle *et al.* constructed adaptively secure FE from differing-input obfuscation (diO), which is a stronger variant of iO [BCP14]. Waters and Garg *et al.* constructed ones from iO and multi-linear maps, respectively [Wat15, GGHZ16]. Ananth *et al.* presented a generic method of constructing an adaptively secure FE from a selective one [ABSV15].

Multi-input functional encryption (MIFE) is a natural extension of FE, which can handle a function class that takes multiple inputs. Roughly speaking, an owner of a secret key for f can learn the computation result $f(m_1, \ldots, m_\mu)$ from ciphertexts of messages $m_1, \ldots, m_\mu$ for some natural number $\mu \geq 1$. Note that MIFE corresponds to FE when $\mu = 1$. Multi-input functional encryption is first defined by Goldwasser *et al.* [GGG+14]. They also constructed MIFE schemes for all circuits from iO, diO, and virtual black-box obfuscation (VBB). Note that VBB is similar to iO but has a stronger security property. In their schemes, the number of inputs is a priori fixed. Later, Badrinarayanan *et al.* proposed unbounded input MI-FE that supports the computation of functions with unbounded arity [BGJS15].

Functional encryption for Turing machines is another interesting research topic that has been

widely studied [BCP14, IPS15, AS16]. Since the complexity class captured by circuits, i.e., $\mathsf{P/poly}$, are larger than that captured by Turing machines, i.e., P, one may wonder what the purpose of considering FE for Turing machines is. A crucial fact is that $\mathsf{P/poly}$ consists of problems efficiently solved by *a family of* circuits whereas P consists of problems efficiently solved by *a* Turing machine. In FE for circuits, secret keys are associated with a circuit (not a family of circuits) and thus the input length or plaintext size is fixed. On the other hand, in FE for Turing machines, secret keys are associated with a Turing machine and thus the plaintext size is arbitrary. Hence, they are incompatible in the context of FE. Boyle *et al.* and Ishai *et al.* constructed FE for Turing machines from diO [BCP14, IPS15], and Ananth and Sahai constructed one from iO [AS16].

The relationship between iO and FE or relations among various types of FE has recently become one of main topics in FE. These works are important to figure out what kind of FE is sufficient to construct iO or full-fledged FE, since constructing iO and full-fledged FE from standard assumptions is a very important open problem in recent cryptology. That is, if an easy-to-construct FE implies iO or full-fledged FE, the problem of constructing them from standard assumptions is reduced to the problem of constructing the easy FE.

Before moving onto specific works, we introduce two notions on FE. The first is single-key vs. many-key. Single-key FE is IND-CPA secure only when the adversary obtain at most one secret key via a secret-key query, while many-key FE is IND-CPA secure even if the adversary can make an arbitrary number of secret-key queries. The second is succinctness (which is also called compactness). FE is called succinct/sub-linearly succinct if its encryption time depends on the maximum circuit size for key generation logarithmically/sub-linearly.

In this paragraph, FE refers to public-key FE for circuits unless otherwise specified. Ananth *et al.* and Bitansky *et al.* showed that many-key non-succinct FE implies many-key succinct FE [AJS15, BV15]. Ananth *et al.* and Bitansky *et al.* showed that single-key sub-linearly succinct FE implies iO with sub-exponential security loss [AJ15, BV15]. Li *et al.* and Garg *et al.* showed that single-key sub-linearly succinct FE implies many-key sub-linearly succinct FE [LM16, GS16]. These three implications are for selectively secure FE. Kitagawa *et al.* showed that selectively secure singe-key sub-linearly succinct FE implies full-fledged FE, namely, adaptively secure many-key succinct FE [KNTY19]. Kitagawa *et al.* showed private-key many-key non-succinct FE also implies iO [KNT18].

Inner Product Functional Encryption. Inner product functional encryption (IPFE), the subject of this thesis, is FE that supports inner products as a function class. In an IPFE scheme, a message to be encrypted is a vector, and a function to be associated with a secret key is also specified by a vector. Decryption of the ciphertext for $\mathbf{x}$ with the secret key for $\mathbf{y}$ reveals the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ and nothing else. Inner product functional encryption allows us to encrypt, for instance, a numerical database to store it in an untrusted server (adversary) and compute linear functions such as weighted means over the encrypted database without revealing any additional information to the server. We present the background on IPFE in Chapter 2.

We also consider MIFE for inner products in this thesis, which is called multi-input inner product functional encryption (MIPFE). In an MIPFE scheme, a secret key $\mathsf{sk}_{\mathbf{y}_1,\ldots,\mathbf{y}_\mu}$ is associated with

vectors $\mathbf{y}_1, \ldots, \mathbf{y}_\mu$, and decrypting ciphertexts $\mathsf{ct}_{\mathbf{x}_1}, \ldots, \mathsf{ct}_{\mathbf{x}_\mu}$ of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_\mu$ with $\mathsf{sk}_{\mathbf{y}_1,\ldots,\mathbf{y}_\mu}$ reveals the summation of the inner products $\sum_{i\in[\mu]}\langle \mathbf{x}_i, \mathbf{y}_i\rangle$. When $\mu = 1$, this corresponds to an IPFE scheme.

Chapter 2

Inner Product Functional Encryption

2.1 Inner Product Functional Encryption: Survey

Functional encryption is a useful tool to delegate computation to untrusted servers since it allows the servers to compute functions over encrypted data. Consider an example case where a user, say Alice, who has only a powerless computer wants to perform a computationally heavy analysis on her data, which is intractable for her computer. A possible solution is to delegate the computation to a powerful cloud server. However, the solution will not work if she is unwilling to reveal her entire data to the untrusted server since it contains sensitive information (in this scenario, the adversary is the server). Functional encryption solves this problem, that is, she can encrypt the data before sending it to the server so that the server cannot read the data. When she needs analysis results for her data, she sends a secret key whose function corresponds to the analysis to the server. The server can compute the analysis results by decrypting the encrypted data by the secret key, while the security of FE guarantees that the server does not obtain any information on her data other than the analysis result. Note that if the algorithm of the analysis also contains sensitive information, the function-hiding property is necessary. This is because the property guarantees that the secret key does not reveal the information of the corresponding function.

Although ABE is a subclass of FE, it only provides cryptographic access control for *entire* data. In contrast to more general FE, ABE does not enable us to obtain *processed information* from encrypted data. Thus, ABE cannot be used for delegation of computation to cloud servers like the above scenario. The first such an FE scheme to be achieved was one for general circuits [GGH+13], which uses indistinguishability obfuscation (iO) as a building block. Later, FE for general circuits was also instantiated from multi-linear maps [BLR+15,GGHZ16]. Indistinguishable obfuscation and multi-linear maps are quite strong tools whereas known candidates are quite heavy to implement and have not been well-studied yet. In fact, many candidates were broken, e.g., [CHL+15, CGH+15].

Inner Product Functional Encryption. Motivated by the impracticality of FE for general circuits, Abdalla *et al.* proposed the first simple FE scheme for inner products aiming for an efficient FE scheme based on standard assumptions [ABDP15]. They constructed two selectively secure IPFE schemes, one is based on the DDH assumption and the other is based on the LWE assumption. The inner product is less powerful than general circuits, but it is a still a useful function and can be used in statistical computation, biometrics, and so on [ABDP15, KLM+18]. This is a reason that Abdalla *et al.* focused on inner products.

As mentioned in Section 1.2, however, the selective security model is less desirable than the adaptive security model. Agrawal *et al.* proposed adaptively secure IPFE schemes based on the DDH, LWE, and decisional composite residuosity (DCR) assumptions [ALS16], which improved the security of IPFE schemes. Besides adaptive security, constructing the first IPFE scheme based on the DCR assumptions is another significant contribution of their work. The salient feature of the DCR-based IPFE scheme is that it can handle inner products for modulo p since it uses a group where the discrete logarithm problem is easy. It is in contrast to the DDH-based IPFE scheme, which uses a group where the discrete logarithm problem is hard. Castagnos *et al.* constructed a more efficient and adaptively secure IPFE scheme for modulo p based on a class group [CLT18]. Their scheme is similar to the DCR-based IPFE scheme, that is, it also leverages a group where the discrete logarithm problem is easy, which is induced by a class group.

The security against chosen-ciphertext attack [RS92] for IPFE was studied in [BBL17]. An encryption scheme satisfies indistinguishability under the chosen ciphertext attack (IND-CCA) if the adversary cannot distinguish two ciphertexts even if it has an access to a decryption oracle. Thus, IND-CCA is stronger than IND-CPA. IND-CCA security is considered as a de facto standard in practice, since there are practical attacks not captured by the IND-CPA model [Ble98].

Simulation-based security for IPFE has been considered in [Wee17, AGRW17, ALMT20]. As mentioned in Section 1.2, simulation-based security is impossible to achieve for some function classes. However, inner product is a simple function and not applicable to the impossibility. Wee and Agrawal *et al.* the authors proposed selective or semi-adaptive IPFE schemes based on DDH [Wee17, AGRW17]. Later, Agrawal *et al.* proposed several adaptively secure IPFE schemes based on DDH, LWE, and DCR [ALMT20].

Unboundedness is a well-studied and important notion in the area of ABE, and Tomida and Takashima defined and realized unbounded IPFE [TT18]. Concurrently, Dufour Sans and Pointcheval also presented unbounded IPFE schemes [DP19]. In Section 2.2.2, we describe the background on unboundedness in detail. Tight security is another important notion for cryptographic schemes and has been extensively studied. Tomida proposed the first IPFE schemes with tight security in [Tom19]. In Section 2.2.3, we elaborate on the background on tight security.

Multi-Input Inner Product Functional Encryption. Analogous to IPFE, we could naturally consider whether we can efficiently construct MIFE for inner products. Abdalla *et al.* answered this question and extended IPFE to the multi-input setting and constructed an MIPFE scheme from pairings [AGRW17]. A pairing (in the context of cryptography) is an efficiently computable bilinear map from two points of an elliptic curve to a finite field where some mathematical problem

is hard. They found that MIPFE is trivially constructible from IPFE in the public-key setting, where all users can encrypt messages with public-keys. In contrast, MIPFE in the private-key setting is non-trivial, where only users who have a maser secret key can encrypt messages.

Later, Abdalla *et al.* constructed MIPFE schemes based on DDH, LWE, and DCR [ACF+18]. Technically, they proposed a generic transformation from IPFE to MIPFE. Since IPFE is constructed from DDH, LWE, and DCR, the transformation gives the corresponding MIPFE schemes based on these assumptions. Especially, IPFE is constructible without pairings [ABDP15, ALS16], and thus their MIPFE scheme based on DDH does not use pairings. Group operations in pairing-free groups is much faster than those in pairing-friendly groups. Thus, the construction without pairings is far more efficient than the previous construction from pairings.

Multi-Client Inner Product Functional Encryption. Multi-input functional encryption (MIFE) is useful when a user computes function values from encrypted data derived from multiple owners. In some cases, however, MIFE leaks too much information to the evaluator. Let us consider the following example. Suppose a user U is allowed to analyze data $D_{A,1}$ and D_B owned by owners A and B, respectively, by aggregating these data. Since A and B are not willing to reveal all data to U, they use MIFE for the analysis. Later, U is allowed to analyze data $D_{A,2}$ and D_C owned by owners A and C, respectively, where $D_{A,2}$ is different from $D_{A,1}$. The problem is that U, who has already obtained the ciphertext for $D_{A,1}$, can use $D_{A,1}$ for the latter analysis, since MIFE schemes allow such decryption, while A may not assume that U make an analysis with $D_{A,1}$ and D_C. To eliminate this inconvenience, the notion, multi-client functional encryption (MCFE) was introduced [GGG+14].

In an MCFE scheme, each ciphertext is associated with a label, and decryption is possible only when all ciphertexts to be decrypted have the same label. Thus, possible decryption patterns can be controlled by labels. MCFE subsumes MIFE because MCFE with one label is the same as MIFE.

Multi-client functional encryption for inner product was first proposed in [CDG+18]. They also proposed decentralized MCFE, in which the key generation process is done by not an authority but each encryptor. In this work, they used the random oracle model and one-label restriction. The random oracle model assumes the existence of an oracle that returns a random outputs for each input, which does not exist in the real world. One-label restriction prohibits an adversary from obtaining more than one ciphertext per label, which is an unnatural model. Thus, it is desirable to construct schemes without using these factors.

The random oracle model was removed in [ABKW19], although the ciphertext size is bigger than the previous scheme. The one-label restriction was removed in [ABG19], and they additionally proposed a generic transformation from IPFE to MCFE for inner products. Thus, they obtained MCFE schemes for inner products from DDH, LWE, and DCR. Independently, Libert and Titiu also proposed an MCFE scheme for inner products from LWE via another technique [LT19].

Function-Hiding Inner Product Functional Encryption. The function-hiding property is first considered for FE for general functions [BS15]. Later, Bishop *et al.* studied efficient con-

structions of the function-hiding IPFE [BJK15], since the function-hiding FE scheme for general functions is also inefficient. Specifically, they proposed an efficient IPFE scheme that satisfied weakly function-hiding property, which imposes unnatural restriction on an adversary in the security game.

Datta *et al.* first removed the restriction and constructed fully function-hiding property. However, their scheme is less efficient than that by Bishop *et al.* This is due to their security proof technique, which requires large ciphertexts and secret keys. Tomida *et al.* improved the scheme by Datta *et al.* and proposed a fully function-hiding IPFE scheme that is as efficient as that by Bishop *et al.* [TAO16]. This contribution is explained in Section 2.2.1. Lin also constructed a fully function-hiding IPFE scheme that is as efficient as that by Bishop *et al.* via a different technique from ours [Lin17]. Her scheme is based on the double layered construction [Lin17], while the previous three schemes including ours are based on the dual pairing vector spaces (DPVS) framework by Okamoto and Takashima [OT10]. The most efficient function-hiding scheme is proposed in [KLM+18], which is based on the generic group model. The generic group model is a model where we can use an idealized cyclic group and thus the security proof is basically easy. On the other hand, the security guarantee given in the generic group model is weaker than that in the standard model. They also gave several applications of function-hiding IPFE to, for example, biometrics and its benchmarks of their implementations.

The function-hiding property for MIPFE is also achieved by Datta *et al.* and Abdalla *et al.* [DOT18, ACF+18]. Intuitively, the MIPFE scheme by Datta *et al.* can be seen as extending of the IPFE scheme by Tomida *et al.* [TAO16], and the MIPFE scheme by Abdalla *et al.* can be seen as extending of the IPFE scheme by Lin [Lin17].

Functional Encryption for Quadratic Functions. Functional encryption for inner products is equivalent to FE for linear functions, since the elements of a secret-key vector can be seen as coefficients of the linear function. Efficient FE for quadratic functions from standard assumptions is constructed from pairings by Baltico *et al.* [BCFG17]. Perceptive readers may think that IPFE can compute quadratic functions by encrypting all quadratic terms in advance, and this idea is correct. The important feature of FE for quadratic functions is that the ciphertext is compact, meaning that the size is linear in the number of elements to be encrypted. They proposed two schemes. One is selectively secure in the indistinguishability-based security definition and based on standard assumptions. The other is adaptively secure in the indistinguishability-based security definition, and its security relies on the generic group model.

Later, Gay proposed FE for quadratic functions with stronger security [Gay20]. That is, his scheme is semi-adaptively secure in the simulation-based security definition. The semi-adaptive security is similar but stronger than the selective security

Machine learning has recently become a major subject in computer science, and the demand fro its commercial use is drastically increasing. For privacy reasons, it is desirable for us to be able to use classifiers to encrypted data in many cases. Functional encryption is expected to be an useful tool for such a purpose. Since a classifier can be seen as a function, we can classify encrypted data by a secret key for a classifier without revealing other information on the underlying data.

Ryffel *et al.* proposed an application of FE for quadratic functions to partially hide original data in prediction [RPB+19].

2.2 Contributions

In this section, we present an overview of our contributions in the progress of IPFE.

2.2.1 Efficient Function-Hiding Inner Product Functional Encryption

In [TAO20], which is majorly revised from the conference version [TAO16], we propose an efficient function-hiding IPFE scheme. We present this contribution in Chapter 4.

Prior to our work, Bishop *et al.* took the first step toward a private-key IPFE scheme with the function-hiding property [BJK15]. However, their security definition for the function-hiding property was somewhat unnatural, which is currently called weakly function-hiding. For private-key IPFE, function-hiding security considers an adversary that interacts with an encryption oracle Enc_b and a key generation oracle KeyGen_b, where b is randomly chosen from $\{0,1\}$. The adversary can query the encryption oracle on a pair of messages $(\mathbf{x}_\ell^0, \mathbf{x}_\ell^1)$ and obtain a ciphertext for $\mathbf{x}_\ell^b$. It can also obtain a secret key for $\mathbf{y}_j^b$ from the key generation oracle similarly. Here, ℓ and j refer to the indices of queries to the encryption oracle and key generation oracle, respectively. To make the game meaningful where the adversary's goal is to guess b, the minimum restriction for the adversary is $\langle \mathbf{x}_\ell^0, \mathbf{y}_j^0 \rangle = \langle \mathbf{x}_\ell^1, \mathbf{y}_j^1 \rangle$ for all ℓ and j, as otherwise the adversary can trivially determine b. After the query phase, if no PPT adversaries can guess b with non-negligible advantage, we say that the private-key IPFE scheme is fully function-hiding. On the other hand, the weakly function-hiding security requires the restriction for the adversary to be $\langle \mathbf{x}_\ell^0, \mathbf{y}_j^0 \rangle = \langle \mathbf{x}_\ell^1, \mathbf{y}_j^0 \rangle = \langle \mathbf{x}_\ell^0, \mathbf{y}_j^1 \rangle = \langle \mathbf{x}_\ell^1, \mathbf{y}_j^1 \rangle$. How this additional restriction affects security is unclear. Later, Datta *et al.* constructed a fully function-hiding IPFE scheme, which satisfies a natural security definition [DDM16]. We construct a more efficient private-key IPFE scheme with the fully function-hiding from a weaker assumption than those by Datta *et al.* Note that the weaker assumption the better since it provides a stronger security guarantee.

Assumption. The schemes in [BJK15, DDM16] are secure under the symmetric external Diffie-Hellman (SXDH) assumption, while our scheme is secure under the k-linear (k-Lin) assumption. It is well known that the k-Lin assumption for $k = 1$ corresponds to the SXDH assumption and becomes weak as k increases [EHK+17]. Hence, the security of our scheme is based on the weaker assumption than that used in [BJK15,DDM16], considering the case where $k > 1$. We also mention the XDLIN assumption, which is used in the conference version [TAO16]. The XDLIN assumption is a variant of the DLIN assumption and implied by it. Since the DLIN assumption corresponds to the k-Lin assumption for $k = 2$, the k-Lin assumption is also weaker than the XDLIN assumption for $k > 1$.

Efficiency. Before discussing the efficiency, we recall techniques used in private-key IPFE schemes. The schemes in [BJK15, DDM16] and our scheme are all based on the dual pairing vector spaces

Table 2.1: Comparison of private-key IPFE schemes. A natural number m is a vector length, msk size refers to a number of $\mathbb{Z}_p$ elements, and sk size and ct size refer to numbers of G_1 and G_2 elements, respectively, #Exponentiation refers to the number of exponentiations in encryption and key generation, and #Paring refers to the number of pairings in decryption. When $k = 1$, the k-Lin assumption is identical to the SXDH assumption.

	BJK15 [BJK15]	DDM16 [DDM16]	Our Scheme
Function-hiding	weak	full	full
msk size	$8m^2 + 8$	$8m^2 + 12m + 28$	$4m^2 + (8k + 2)m + 2k(2k + 1)$
ct, sk size	$2m + 2$	$4m + 8$	$2m + 2k + 1$
#Exponentiation	$2m + 2$	$4m + 8$	$2m + 2k + 1$
#Paring	$2m + 2$	$4m + 8$	$2m + 2k + 1$
Assumptions	SXDH	SXDH	k-Lin

(DPVS) framework introduced by Okamoto and Takashima [OT09,OT10], and they have the same structure. That is, a master secret key is orthonormal bases of DPVS, secret keys and ciphertexts are vectors of DPVS, both key generation algorithm and encryption algorithms involve exponentiation, and a decryption algorithm involves pairing operations in bilinear groups. Our scheme is superior to the scheme by Datta et al. with a constant factor 2 in terms of both space and computational efficiency, assuming $k \ll n$ (Table 2.1).

2.2.2 Unbounded Inner Product Functional Encryption

In [TT20], the conference version of which is [TT18], we propose the first unbounded IPFE schemes. We present this contribution in Chapter 5. The motivation of this work is that all IPFE schemes prior to this work have one inconvenient property: they are *bounded*. That is, we need to fix the maximum length of vectors to be handled in the scheme at the beginning. After fixing the maximum length, we cannot handle vectors whose lengths exceed it. This is very inconvenient because it is almost impossible in the setup phase to predict which data will be encrypted. One may think that we can solve the problem by setting the maximum length to a quite large value. However, the size of a public parameter of bounded schemes expands at least linearly with the fixed maximum length, and such a solution incurs an unnecessary efficiency loss. Hence, it is desirable that we do not need to declare the maximum length of vectors to be handled in the scheme at the beginning and can make encryption or key generation for vectors with unbounded lengths. In the context of inner product encryption (IPE) [KSW08] and attribute-based encryption [GPSW06], there exist unbounded schemes [LW11,OT12b,BV16,CGKW18], whose public parameters do not impose a limit on the maximum length of vectors or number of attributes used in the scheme. Thus, we naturally have the following question:

Can we construct IPFE schemes that can handle vectors with unbounded lengths?

Table 2.2: Comparison among private-key schemes that are fully function-hiding and public-key schemes with adaptive security in the standard model. The schemes refer to DDM16 [DDM16], TAO16 [TAO16], KKS17 [KKS17], and ALS16 [ALS16]. Although Lin also presented a construction of function-hiding scheme [Lin17], her scheme is the selective secure one and we do not adopt it here. A natural number $m \in \mathbb{N}$ denotes a length of a vector associated with the ciphertext or secret key. In our schemes, α denotes a bit length that is necessary to specify an index set associated with a vector. In the ALS16 scheme, β denotes a bit length that is necessary to specify a vector to be embed into a secret key. In this table, we omit a group description in a public key.

private-key scheme					
scheme	$\lvert\mathsf{msk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{sk}\rvert$	pairing	assump.
DDM16	$(8m^2+12m+28)\lvert\mathbb{Z}_p\rvert$	$(4m+8)\lvert G_1\rvert$	$(4m+8)\lvert G_2\rvert$	Yes	SXDH
TAO16	$(4m^2+18m+20)\lvert\mathbb{Z}_p\rvert$	$(2m+5)\lvert G_1\rvert$	$(2m+5)\lvert G_2\rvert$	Yes	XDLIN
KKS17	$(6m+8)\lvert\mathbb{Z}_p\rvert$	$(2m+8)\lvert G_1\rvert$	$(2m+8)\lvert G_2\rvert$	Yes	SXDH
Ours 1	$\lvert$PRF key$\rvert$	$4m\lvert G_1\rvert$	$4m\lvert G_2\rvert+\alpha$	Yes	SXDH

public-key scheme						
scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{msk}\rvert$	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{sk}\rvert$	pairing	assump.
ALS16	$(m+1)\lvert G\rvert$	$2m\lvert\mathbb{Z}_p\rvert$	$(m+2)\lvert G\rvert$	$2\lvert\mathbb{Z}_p\rvert+\beta$	No	DDH
Ours 2	$28\lvert G_1\rvert$	$28\lvert\mathbb{Z}_p\rvert$	$7m\lvert G_1\rvert$	$7m\lvert G_2\rvert+\alpha$	Yes	SXDH

Our contributions. We answer the question affirmatively. More precisely, we construct two concrete unbounded IPFE (UIPFE) schemes on the basis of the standard SXDH assumption that are both secure in the standard model.

1. The first scheme is private-key IPFE with the fully function-hiding property, which is the strongest indistinguishability-based security notion when considering function privacy [DDM16].

2. The second scheme is public-key IPFE with adaptive security, which is a standard and desirable indistinguishability-based security notion [ALS16].

Table 2.2 compares efficiency among private-key schemes that are fully function-hiding and public-key schemes with adaptive security in the standard model. Both our schemes achieve almost the same efficiency as the previous bounded fully function-hiding IPFE schemes except the small constant factor. Note that previous public-key based schemes do not need pairing when instantiated from a cyclic group [ABDP15, ALS16]. However, we do not know how to construct unbounded public-key based IPFE schemes *without pairing*.

In UIPFE schemes, we can consider various conditions about encryption, key generation, and decryption. It is another important merit of UIPFE. For encryption and key generation, we can consider two cases, *consecutive* and *separate*. In the consecutive setting, each element of a vector is automatically indexed to its position when the vector is input to an encryption or key generation algorithm, i.e., for a vector (a, b, c), a's index is set to 1, b's index to 2, and c's index to 3. On the other hand, in the separate setting, an index set is attached to a vector and encryption and key

generation are executed correspondingly to its index set. In other words, a vector (a, b, c) is indexed by some set, e.g., $\{1, 5, 6\}$, and the indices of a, b and c are set to 1,5, and 6, respectively. A separate scheme obviously implies a consecutive scheme with respect to encryption or key generation. Next, we focus on the conditions of decryption. Similar to [OT12b], we can classify the decryptable condition of IPFE schemes into three types: *ct-dominant*, *sk-dominant*, and *equal*. Let S_{ct} be an index set of a ciphertext ct and S_{sk} be an index set of a secret key sk. Then ct is decryptable with sk iff $S_{\mathsf{ct}} \supseteq S_{\mathsf{sk}}$ in ct-dominant schemes, $S_{\mathsf{ct}} \subseteq S_{\mathsf{sk}}$ in sk-dominant schemes, and $S_{\mathsf{ct}} = S_{\mathsf{sk}}$ in equal schemes. We denote the type of the schemes described above as (E:xx, K:yy, D:zz) where xx, yy $\in$ {con, sep}, and zz $\in$ {ct-dom, sk-dom, eq}, which means that encryption is xx setting, key generation is yy setting, and decryption is zz setting. It is not difficult to observe that the setting (E:sep, K:con, D:ct-dom) is meaningless because only the *consecutive* part of *separate* ciphertexts can be decrypted with any *consecutive* secret key. For example, for a ciphertext with an index set $\{1, 2, 4\}$, the element indexed as 4 is never used for decryption in the K:con setting. Hence, it is the same as the (E:con, K:con, D:ct-dom) setting. Similarly, (E:con, K:sep, D:sk-dom), (E:con, K:sep, D:eq), and (E:sep, K:con, D:eq) are also meaningless. Thus, we can consider eight types of UIPFE schemes.

In this paper, we focus on the D:ct-dom setting because we believe it is the most convenient for real applications. Consider the situation where Alice holds a huge encrypted database in an untrusted server. When she wants the server to make some computation over the database, she can obtain the result by sending a corresponding secret key to the server. If the necessary part of the database for the computation is very small, the D:ct-dom setting allows Alice to issue a compact secret key. This is because the size of a secret key of IPFE schemes typically grows linearly to the length of the corresponding vector. In the other settings, Alice needs to issue a secret key that is at least larger than some constant multiple of the size of the database, and this incurs a big efficiency loss.

Both our schemes are the (E:con, K:sep, D:ct-dom) setting, which implies (E:con, K:con, D:ct-dom). Some readers may wonder why we do not consider the most general setting of D:ct-dom, (E:sep, K:sep, D:ct-dom), which implies all D:ct-dom schemes. The reason is we can prove the security of our schemes against adaptive adversaries only in the (E:con, K:sep, D:ct-dom) setting. The intuitive reason for this limitation is that, in security proofs, reduction algorithms need to guess the contents of an index set with which an adversary queries an encryption oracle. This is possible in the E:con setting because the length of vectors queried by an adversary is a polynomial and a reduction algorithm can correctly guess the length with a non-negligible probability. In the E:sep setting, however, the possibility of index sets is exponential and is unpredictable for reduction algorithms. For this reason, our schemes are secure against selective adversaries in the (E:sep, K:sep, D:ct-dom) setting. In particular, our public-key scheme is semi-adaptively secure in the (E:sep, K:sep, D:ct-dom) setting, which means that the adversary declares a challenge message right after obtaining a public key in a security game [CW14]. Note that the fully function-hiding private-key IPFE scheme in the (E:sep, K:con, D:sk-dom) setting is trivial with our scheme because the roles of ciphertexts and secret keys are the same in fully function-hiding private-key IPFE. In addition, the fully function-hiding private-key IPFE in the (E:sep, K:sep, D:eq) setting is easily

Table 2.3: Summary of our result. The asterisk indicates that the scheme is not explicitly demonstrated in this paper, but trivially implied by our result. A symbol $\perp$ indicates that the scheme is meaningless.

private-key scheme				
	E:con, K:con	E:sep, K:con	E:con, K:sep	E:sep, K:sep
D:ct-dom	full*	$\perp$	full (Sec.5.2.1)	selective (Sec.5.2.3)
D:sk-dom	full*	full*	$\perp$	selective*
D:eq	full*	$\perp$	$\perp$	full (Sec.5.2.4)
public-key scheme				
	E:con, K:con	E:sep, K:con	E:con, K:sep	E:sep, K:sep
D:ct-dom	adaptive*	$\perp$	adaptive (Sec.5.3.1)	semi-adaptive (Sec.5.3.4)
D:sk-dom	open	open	$\perp$	open
D:eq	open	$\perp$	$\perp$	open

constructible. We summarize our result in Table 2.3.

Concurrent Work. Concurrently with our work [TT18], Dufour Sans and Pointcheval also presented UIPFE schemes [DP19]. In our terminology, they proposed public-key (E:sep, K:sep, D:eq) and (E:sep, K:sep, D:ct-dom) schemes in their paper. Their schemes have short secret keys, meaning that they contain one group element and a corresponding vector. However, their schemes rely on the random oracle model and achieve only the selective security, and their (E:sep, K:sep, D:ct-dom) scheme also relies on a new interactive assumption. More precisely, they assume that a kind of problem is hard for all PPT adversaries even if they are allowed to access some oracles. In addition, their (E:sep, K:sep, D:ct-dom) scheme does not have collusion resistance of illegitimate secret keys, which means that a combination of illegitimate keys can become a legitimate key.

2.2.3 Tightly Secure Inner Product Functional Encryption

In [Tom20], the conference version of which is [Tom19], we study tight security of IPFE and propose the first tightly secure IPFE scheme including function-hiding and multi-input schemes. We present this contribution in Chapter 6.

Tight security. As explained in Chapter 1, when we try to prove the security of a cryptographic scheme, we typically 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 if we require the scheme to have (t, ϵ)-security, which means that no probabilistic algorithms can break the scheme with probability more than ϵ within t steps, we need the assumption that no algorithms can break the underlying problem with probability more than ϵ/L within about t steps. Thus, the assumption becomes weaker and more desirable as ϵ/L becomes larger, which means that the smaller L the better.

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 [BBM00]. 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 not so affected by those numbers. We often say that the security reduction is tight if the security loss is constant, i.e., $L = O(1)$, and almost tight if $L = O(\lambda)$ or $L = O(\log q)$ where λ is a security parameter and q is the number of queries made by an adversary.

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 signatures, e.g, [HJ12, CW13, LPJY15, AHY15, GHKW16, Hof16, Hof17, GHK17]. 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. [GKW18]. 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. [AHN+17], 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 [BCM+15, EM14].

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 [Gen]. 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: $m = 4 \times 2^{13} = 2^{15}$ 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.

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 two first tightly secure public-key IPFE schemes in the multi-user and multi-challenge setting. Note that previous IPFE schemes are tightly reduced to underlying assumptions in the single-challenge setting [ALS16], 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 schemes are 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. The security of the first scheme is reduced to the MDDH assumption with a small-constant security loss. The second scheme has smaller ciphertexts and secret keys than the first one, while its security loss depends on the vector length of the scheme (but does not on the number of ciphertexts that an adversary obtains). Thus, the second scheme is effective if the vector length is small.

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 [AGRW17, ACF+18]. 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 schemes satisfy the requirement for the conversion. Thus, we can obtain the first tightly secure MIPFE schemes.

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 [BJK15, DDM16, TAO16, Lin17]. 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 [Lin17]. Applying her paradigm to our IPFE schemes, we can obtain the first tightly secure function-hiding IPFE schemes that are based on bilinear groups. However, the naive application of her paradigm to our schemes results in redundant schemes. Thus, we optimize the schemes 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 [ACF+18]. Furthermore, the techniques used in the rather specific constructions of known function-hiding MIPFE schemes [DOT18, ACF+18] are not applicable to our situation. Roughly speaking, this is because our schemes require the selective setting in a certain step of the proof, if we naively try to prove the security similarly to [DOT18, ACF+18].

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. [ACF+18]: 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 [DOT18, ACF+18] by utilizing our conversion to function-hiding IPFE schemes, e.g., our scheme in Chapter 4. Applying our conversion to our tightly secure function-hiding IPFE schemes, we can finally achieve the first tightly secure function-hiding MIPFE schemes.

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 [ABDP15, KLM+18], 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 [ALS16] or based on a non-standard assumption [CLT18].

We summarize the comparison of our schemes with previous ones in Tables 2.4 to 2.7. 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 first function-hiding IPFE scheme from the SXDH, it takes almost 5 times more pairings in decryption than the state-of-the-art scheme (Table 2.6). 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.

2.3 Organization

We show the organization of the rest of this thesis. In Chapter 3, we introduce basic cryptographic tools and definitions necessary to present our contributions. In Chapters 4 to 6, we present our contributions. We conclude our thesis and show some open problems related to IPFE in Chapter 7.

Table 2.4: 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 k refers to the parameter of the MDDH assumption. The number q_{ct} refers to the total number of ciphertext queries by an adversary. Note that we omit the group description from $|\mathsf{pk}|$.

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

Table 2.5: 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 k refers to the parameter of the MDDH assumption. 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.

MIPFE schemes					
scheme	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	sec. loss	assumption
ACFGU18 [ACF+18]	$\{k^2+k+(k+2)m\}\mu$	$m+k+1$	$(m+k+1)\mu+1$	$O(q_{\mathsf{ct}})$	$\mathcal{D}_k$-MDDH
Ours 1	$(k^2m+km+1)m\mu$	$(k^2+k+1)m$	$(k^2+k+1)m\mu+1$	$O(1)$	$\mathcal{D}_k$-MDDH
Ours 2	$(km+2)m\mu$	$(k+1)m+k$	$\{(k+1)m+k\}\mu+1$	$O(m^2)$	$\mathcal{D}_k$-MDDH

Table 2.6: Comparison of fully function-hiding IPFE schemes in the standard model. Lin17 [Lin17] refers to the scheme obtained by applying her paradigm to the IPFE scheme AGRW17 [AGRW17]. 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 k refers to the parameter of the MDDH assumption. 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 IPFE schemes					
scheme	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	sec. loss	assumption
DDM16 [DDM16]	$8m^2+12m+28$	$4m+8$	$4m+8$	$O(q_{\mathsf{ct}}q_{\mathsf{sk}})$	SXDH
TAO16 [TAO16]	$4m^2+18m+20$	$2m+5$	$2m+5$	$O(q_{\mathsf{ct}}+q_{\mathsf{sk}})$	XDLIN
Lin17 [Lin17]	$(k+1)(4m+3k+1)$	$2m+2k+2$	$2m+2k+2$	$O(q_{\mathsf{ct}}+q_{\mathsf{sk}})$	$\mathcal{D}_k$-MDDH
Ours 1	$32m^2$	$10m$	$10m$	$O(1)$	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
Ours 2	$12m^2+8m+1$	$6m+2$	$6m+2$	$O(m^2)$	SXDH
	$(4k^2+8k)m^2+(4k^2+4k)m+k^2$	$(4k+2)m+2k$	$(4k+2)m+2k$	$O(m^2)$	$\mathcal{D}_k$-MDDH

Table 2.7: 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 k refers to the parameter of the MDDH assumption. 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.

function-hiding MIPFE schemes			
scheme	$\|\mathsf{msk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$
DOT18 [DOT18]	$(2m+2k+1)^2\mu$	$2m+2k+1$	$(2m+2k+1)\mu$
ACFGU18 [ACF+18]	$\{(k+1)(4m+5k+1)+k\}\mu$	$2m+3k+2$	$(2m+3k+2)\mu(+\|G_T\|)$
Ours 1	$\{(k^4+2k^3+3k^2+2k)(2m+1)^2+m\}\mu$	$(2k^2+2k+1)(2m+1)$	$(2k^2+2k+1)(2m+1)\mu$
Ours 2	$\{(k^2+2k)(2m+1)^2+(2k^2+2k)(2m+1)+k^2+m\}\mu$	$(4k+2)m+3k+1$	$\{(4k+2)m+3k+1\}\mu$
scheme	sec. loss	assumption	
DOT18 [DOT18]	$O(q_{\mathsf{ct}}+q_{\mathsf{sk}})$	k-Lin	
ACFGU18 [ACF+18]	$O(q_{\mathsf{ct}}+\mu q_{\mathsf{sk}})$	$\mathcal{D}_k$-MDDH	
Ours 1	$O(m^2)$	$\mathcal{D}_k$-MDDH	
Ours 2	$O(m^2)$	$\mathcal{D}_k$-MDDH	

Chapter 3

Preliminaries

In this chapter, we introduce basic notions and definitions that are used to present our contributions.

3.1 Notations

We denote the string $0, \ldots, 0$ where the number of 0 is n by 0^n and $1, \ldots, 1$ by 1^n similarly. For a natural number $n \in \mathbb{N}$, $\mathbb{Z}_n$ denotes a ring $\mathbb{Z}/n\mathbb{Z}$ and $[n]$ denotes a set $\{1, \ldots, n\}$. For natural numbers $n, m \in \mathbb{N}$, $[n]$ denotes a set $\{1, \ldots, n\}$, and $[m, n]$ denotes a set $\{m, \ldots, n\}$ (if $m > n, [m, n] := \phi$). For a set S, $s \leftarrow S$ denotes that s is uniformly chosen from S. For a random variable D, $d \leftarrow D$ denotes that d is chosen following the distribution of D. For a vector $\mathbf{x}$, $||\mathbf{x}||_\infty$ denotes its infinity norm. For vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ denotes a vector generated by the vertical concatenation of these vectors. For matrices (including vectors) with the same number of rows $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$, $(\mathbf{A}_1||\mathbf{A}_2|| \cdots ||\mathbf{A}_n)$ denotes a matrix generated by the horizontal concatenation of these matrices. For a field K, $\mathsf{M}_n(K)$ and $\mathsf{GL}_n(K)$ denote a set of all $n \times n$ matrices and all $n \times n$ regular matrices whose elements are in K, respectively. We use a bold upper-case letter to denote a matrix, e.g., $\mathbf{A}$, and a bold lower-case version of the same letter with subscript i to denote the i-th row of the matrix, e.g., $\mathbf{a}_i$. For example, $\mathbf{a}_i$ denotes the i-th row of $\mathbf{A}$. For a regular matrix $\mathbf{A}$, $\mathbf{A}^*$ denotes $(\mathbf{A}^{-1})^\top$. 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. For a generator g_ι of a cyclic group G_ι, a matrix $\mathbf{A}$, and vector $\mathbf{a}$, $[\mathbf{A}]_\iota$ and $[\mathbf{a}]_\iota$ denote the corresponding matrix and vector on the exponent of g_ι, respectively. For vectors $\mathbf{x} := (x_1, \ldots, x_n)$ and $\mathbf{y} := (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, 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 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 $Y := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$, we say that X and Y are computationally indistinguishable if, for all PPT adversaries $\mathcal{A}$ and $\lambda \in \mathbb{N}$, we have $|\Pr[1 \leftarrow \mathcal{A}(1^\lambda, x) \mid x \leftarrow X_\lambda] - \Pr[1 \leftarrow \mathcal{A}(1^\lambda, y) \mid y \leftarrow Y_\lambda]| \leq \mathsf{negl}(\lambda)$. $X \approx_c Y$ means that they are computationally indistinguishable.

3.2 Basic Tools and Assumptions

In this section, we define basic cryptographic tools and assumptions that our schemes presented in this thesis are based on.

Definition 3.1 (Pseudorandom Functions)**.** A pseudorandom function (PRF) family $\mathcal{F}:=\{F_K\}_{K\in\mathcal{K}_\lambda}$ with a key space $\mathcal{K}_\lambda$, a domain $\mathcal{X}_\lambda$, and a range $\mathcal{Y}_\lambda$ is a function family that consists of functions $F_K : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. Let $\mathcal{R}_\lambda$ be a set of functions consisting of all functions whose domain and range are $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ respectively. For any PPT adversary $\mathcal{A}$, the following condition holds,

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(\lambda) := \left|\mathsf{Pr}[1 \leftarrow \mathcal{A}^{F_K(\cdot)}] - \mathsf{Pr}[1 \leftarrow \mathcal{A}^{R(\cdot)}]\right| \leq \mathsf{negl}(\lambda),$$

where $K \leftarrow \mathcal{K}_\lambda$ and $R \leftarrow \mathcal{R}_\lambda$.

Definition 3.2 (Cyclic Group)**.** A description of a cyclic group $\mathbb{G}_{\mathsf{CG}}:=(p, G, g)$ consists of a prime p, a cyclic group G of order p, and a generator g. A cyclic group generator $\mathcal{G}_{\mathsf{CG}}(1^\lambda)$ takes security parameter 1^λ and outputs a description of a cyclic group $\mathbb{G}_{\mathsf{CG}}$ with a λ-bit prime p.

Definition 3.3 (Bilinear Groups)**.** A description of bilinear groups $\mathbb{G}_{\mathsf{BG}}:=(p, G_1, G_2, G_T, g_1, g_2, e)$ 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 $\mathcal{G}_{\mathsf{BG}}(1^\lambda)$ takes security parameter 1^λ and outputs a description of bilinear groups $\mathbb{G}_{\mathsf{BG}}$ with a λ-bit prime p.

Definition 3.4 ($\mathcal{D}_k$-MDDH Assumption [EHK+17])**.** 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:

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

We say that the $\mathcal{D}_k$-matrix decisional Diffie-Hellman (MDDH) assumption holds with respect to $\mathcal{G}_{\mathsf{CG}}$ if, for any PPT adversary $\mathcal{A}$,

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

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

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

We denote $\max_i \mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{A},\mathsf{BG},i}(\lambda)$ by $\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{A}}(\lambda)$.

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:

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

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

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

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

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

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

k-Lin Assumption. The k-Lin assumption is well-known as a concrete instance of the MDDH assumption. For $k \in \mathbb{N}, a_1, \ldots, a_k \in \mathbb{Z}_p$, the k-Lin assumption is define as the $\mathcal{D}_k$-MDDH assumption with $\mathcal{D}_k$ being the uniform distribution of matrices with the following form:

$$\mathbf{A} = \begin{pmatrix} 1 & \cdots & 1 \\ a_1 & & \\ & \ddots & \\ & & a_k \end{pmatrix} \in \mathbb{Z}_p^{(k+1)\times k}$$

Especially, the 1-Lin assumption (the k-Lin assumption where $k = 1$) is called the symmetric external Diffie-Hellman (SXDH) assumption in bilinear groups. Observe that the SXDH assumption says that $(\mathbb{G}, [a], [v], [av]) \approx_c (\mathbb{G}, [a], [v], [t])$ where $a, v, t \leftarrow \mathbb{Z}_p$. We use the SXDH assumption in this form in Chapter 5.

Definition 3.5 (Dual Pairing Vector Spaces [OT10])**.** For a natural number $n \in \mathbb{N}$, we choose random dual orthonormal bases $(\mathbf{B}, \mathbf{B}^*)$ as $\mathbf{B} \leftarrow \mathsf{GL}_n(\mathbb{Z}_p)$ and $\mathbf{B}^* := (\mathbf{B}^{-1})^\top$. Then $[\mathbf{B}]_1$ and $[\mathbf{B}^*]_2$ are dual orthonormal bases of vector spaces $V := G_1^n$ and $V^* := G_2^n$ respectively. Observe that the following two properties hold.

1. For any vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^n$, $e([\mathbf{xB}]_1, [\mathbf{yB^*}]_2) = e(g_1, g_2)^{\langle \mathbf{x}, \mathbf{y}\rangle}$.
2. For any vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}_1, \ldots, \mathbf{y}_\ell \in \mathbb{Z}_p^n$ and any matrix $\mathbf{M} \in \mathsf{GL}_n(\mathbb{Z}_p)$, $(\{\mathbf{x}_i\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{B}^*\}_{i\in[\ell]})$ and $(\{\mathbf{x}_i\mathbf{MB}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{B}^*\}_{i\in[\ell]})$ are identically distributed. More generally, for any set $S \subseteq [n]$ s.t. $\forall i \in S, \mathbf{b}_i = \mathbf{M}^{-1}\mathbf{b}_i$, $(\{\mathbf{b}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{B}^*\}_{i\in[\ell]})$

and $(\{\mathbf{b}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{MB}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{B}^*\}_{i\in[\ell]})$ are also identically distributed. This is because $(\mathbf{D}, \mathbf{D}^*) := (\mathbf{M}^{-1}\mathbf{B}, \mathbf{M}^\top\mathbf{B}^*)$ are also random dual orthonormal bases and $(\{\mathbf{b}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{B}^*\}_{i\in[\ell]}) = (\{\mathbf{d}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{MD}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{D}^*\}_{i\in[\ell]})$.

In Chapter 5, we use the fact that randomly chosen matrices whose elements are in $\mathbb{Z}_p$ have a high probability of being invertible. More concretely, we have the following lemma for DPVS, which is implicitly shown in [DOT18].

Lemma 3.1. *Let p be a $\Omega\lambda$-bit prime. For any polynomial $m := m(\lambda)$ and $n := n(\lambda)$, we have*

$$\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | \mathbf{B}_1, \ldots, \mathbf{B}_m \leftarrow \mathsf{M}_n(\mathbb{Z}_p)] = 2^{-\Omega(\lambda)}.$$

3.3 Functional Encryption

In this section, we define functional encryption. Specifically, we define public-key/private-key functional encryption and private-key multi-input functional encryption. Note that we only consider private-key schemes for multi-input functional encryption in this thesis.

3.3.1 Definitions for Functional Encryption

Multi-User and Multi-Challenge. We first clarify the two settings that are related to security models of cryptographic schemes, namely, multi-user and multi-challenge. The multi-user setting is a security model where an adversary can attack multiple instances of a cryptographic scheme. The multi-challenge setting is a security model where an adversary can attack multiple ciphertexts of a cryptographic scheme. In the real world, it is natural to assume that adversaries obtain multiple ciphertexts generated by multiple users. Hence, we should consider the security of cryptographic schemes in the multi-user and multi-challenge setting.

It is known that a public-key cryptographic scheme is basically secure in the multi-user and multi-challenge setting if it is secure in the single-user and single-challenge setting [BBM00]. Similarly, it is known that a private-key cryptographic scheme is basically secure in the multi-user and multi-challenge setting if it is secure in the single-user and multi-challenge setting. Therefore, we normally consider the single-user and single-challenge setting for public-key schemes and the single-user and multi-challenge setting for private-key schemes. However, it is also known that these implications require amplification of security loss [BBM00]. Thus, when we consider tight security, we need to directly handle the multi-user and multi-challenge setting.

In this section, we define security definitions in the multi-user and multi-challenge setting, since this thesis contains the chapter where we consider tight security (Chapter 6). On the other hand, we consider the single-user (and single-challenge for public-key FE) setting for the schemes of which we do not care about the security loss (Chapters 4 and 5). Note that we can treat the single-XX setting as a special case of the multi-XX setting.

Definition 3.6 (Public-Key Functional Encryption)**.** Let $\mathcal{X}$ be a domain and $\mathcal{F}$ be a function class. Public-key functional encryption (Pub-FE) for $\mathcal{F}$ consists of five algorithms.

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

$\mathsf{Setup}(\mathsf{pp})$**:** It takes pp and outputs a public key pk and a master secret key msk.

$\mathsf{Enc}(\mathsf{pk}, x)$**:** It takes pk and $x \in \mathcal{X}$ and outputs a ciphertext ct.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, f)$**:** It takes $\mathsf{pk}, \mathsf{msk}$, and $f \in \mathcal{F}$ 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 or a symbol $\perp$.

This syntax is one for the multi-user setting. In the single-user setting, we can combine Par and Setup into Setup, which takes 1^λ and outputs $\mathsf{pk}, \mathsf{msk}$. This also goes for a private-key scheme (Definition 3.7).

Correctness. Pub-FE is *correct* if it satisfies the following condition. For all $\lambda \in \mathbb{N}, x \in \mathcal{X}, f \in \mathcal{F}$, we have

$$\mathsf{Pr}\left[d = f(x) \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, f) \\ d := \mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

Security. Let $\mu \in \mathbb{N}$ be a natural number that represents the number of users. Pub-FE 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-FE defined as follows is negligible in λ for any constant $\mu \in \mathbb{N}$, and PPT adversary $\mathcal{A}$,

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

The description of the oracles $\mathcal{O}_{\mathsf{ct}}$ and $\mathcal{O}_{\mathsf{sk}}$ is presented in Fig 3.1. 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 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

$$f_{i,\ell_i}(x^0_{i,j_i}) = f_{i,\ell_i}(x^1_{i,j_i}). \tag{3.1}$$

$\underline{\mathcal{O}_{\mathsf{ct}}(\beta \in \{0,1\}, i \in [\mu], (x^0, x^1) \in \mathcal{X}^2)}$	$\underline{\mathcal{O}_{\mathsf{sk}}(i \in [\mu], f \in \mathcal{F})}$
$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}_i, x^\beta)$	$\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{pk}_i, \mathsf{msk}_i, f)$
return ct_i	return sk_i

Fig 3.1: The description of oracles in the security game for Pub-FE.

We can define semi-adaptive security similarly to adaptive security. We impose an additional condition on $\mathcal{A}$, that is, $\mathcal{A}$ cannot query $\mathcal{O}_{\mathsf{ct}}$ after querying $\mathcal{O}_{\mathsf{sk}}$ even once. Pub-FE is *semi-adaptively secure* if the advantage of $\mathcal{A}$ that has the additional condition is negligible for all $\mathcal{A}$.

We can also define the adaptive or semi-adaptive security in the *single-user and single-challenge setting* similarly. That is, $\mu = 1$, and the adversary can query $\mathcal{O}_{\mathsf{ct}}$ at most once in this setting.

Definition 3.7 (Private-Key Functional Encryption)**.** Let $\mathcal{X}$ be a domain and $\mathcal{F}$ be a function class. Private-key functional encryption (Priv-FE) consists of five algorithms.

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

$\mathsf{Setup}(\mathsf{pp})$**:** It takes pp and outputs a master secret key msk.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, x)$**:** It takes pk and $x \in \mathcal{X}$ and outputs a ciphertext ct.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f)$**:** It takes $\mathsf{pp}, \mathsf{msk}$, and $f \in \mathcal{F}$ 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 or a symbol $\perp$.

Correctness. Priv-FE is *correct* if it satisfies the following condition. For all $\lambda \in \mathbb{N}, x \in \mathcal{X}, f \in \mathcal{F}$, we have

$$\mathsf{Pr}\left[d = f(x) \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, f) \\ d := \mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk}) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

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

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

The description of the oracles $\mathcal{O}_{\mathsf{ct}}$ and $\mathcal{O}_{\mathsf{sk}}$ is presented in Fig 3.2. 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

$\underline{\mathcal{O}_{\mathsf{ct}}(\beta \in \{0,1\}, i \in [\mu], (x^0, x^1) \in \mathcal{X}^2)}$	$\underline{\mathcal{O}_{\mathsf{sk}}(\beta \in \{0,1\}, i \in [\mu], (f^0, f^1) \in \mathcal{F}^2)}$
$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}_i, x^\beta)$	$\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}_i, f^\beta)$
return ct_i	return sk_i

Fig 3.2: The description of oracles in the security game for Priv-FE.

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

$$f^0_{i,\ell_i}(x^0_{i,j_i}) = f^1_{i,\ell_i}(x^1_{i,j_i}). \tag{3.2}$$

Similarly to the public-key scheme, we say the scheme is fully function-hiding in the *single-user setting* when $\mu = 1$.

We say that Priv-FE 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

$$f^0_{i,\ell_i}(x^0_{i,j_i}) = f^0_{i,\ell_i}(x^1_{i,j_i}) = f^1_{i,\ell_i}(x^1_{i,j_i}). \tag{3.3}$$

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

We can define the selective security similarly to the fully function-hiding security. We impose an additional condition on $\mathcal{A}$, that is, $\mathcal{A}$ cannot query $\mathcal{O}_{\mathsf{ct}}$ after querying $\mathcal{O}_{\mathsf{sk}}$ even once. Priv-FE is *selectively function-hiding* if the advantage of $\mathcal{A}$ that has the additional condition is negligible for all $\mathcal{A}$.

Definition 3.8 (Private-Key Multi-Input Functional Encryption)**.** Let $\mathcal{X}_1, \ldots, \mathcal{X}_\mu$ be domains and $\mathcal{F}$ be a function class. Private-key multi-input functional encryption (MIFE) consists of four algorithms.

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

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, x_i)$**:** It takes pp, msk, an index $i \in [\mu]$, and $x_i \in \mathcal{X}_i$ and outputs a ciphertext ct_i.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f)$**:** It takes $\mathsf{pp}, \mathsf{msk}$, and $f \in \mathcal{F}$, 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 or a symbol $\perp$.

In what follows, we refer to private-key MIFE just as MIFE.

Correctness. MIFE is *correct* if it satisfies the following condition. For any $\lambda, \mu \in \mathbb{N}$, we have

$$\mathsf{Pr}\left[d = f(x_1, \ldots, x_\mu) \;\middle|\; \begin{array}{l} \mathsf{pp}, \mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda, 1^\mu) \\ \mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, x_i) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f) \\ d := \mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\mu, \mathsf{sk}) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

$\underline{\mathcal{O}_{\mathsf{ct}}(\beta \in \{0,1\}, i \in [\mu], (x_i^0, x_i^1) \in \mathcal{X}_i^2)}$	$\underline{\mathcal{O}_{\mathsf{sk}}(\beta \in \{0,1\}, (f^0, f^1) \in \mathcal{F}^2)}$
$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, x_i^\beta)$	$\mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f^\beta)$
return ct_i	return sk

Fig 3.3: The description of oracles in the security game for MIFE.

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

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

The description of the oracles $\mathcal{O}_{\mathsf{ct}}$ and $\mathcal{O}_{\mathsf{sk}}$ is presented in Fig 3.3. 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 $\ell \in [q_{\mathsf{sk}}]$,

$$f^0_\ell(x^0_{1,j_1}, \ldots, x^0_{\mu.j_\mu}) = f^1_\ell(x^1_{1,j_1}, \ldots, x^1_{\mu.j_\mu}). \tag{3.4}$$

In this thesis, 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 [AGRW17, DOT18].

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

$$f^0_\ell = f^1_\ell.$$

We denote the advantage of $\mathcal{A}$ in the adaptive-security game by $\mathsf{Adv}^{\mathsf{MIFE}}_{\mathcal{A},\mathsf{ad}}(\lambda)$. This security definition captures only the message privacy of MIFE 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 [AGRW17, ACF+18].

3.3.2 Function Classes for Inner Products

In this thesis, we handle various kinds of function classes related to inner products. We formally define them in the following. First, we define inner products over $\mathbb{Z}$ and its extension to the multi-input case, which are related to Chapters 4 and 6. We need to bound infinity norms of vectors in all classes used for correctness of our schemes. This is because decryption algorithms needs to solve the discrete logarithm problem on the decryption value. Note that this is common in all previous IPFE schemes based on a cyclic group or bilinear groups. As pointed out in [KLM+18], however, this step does not affect efficiency so much in many practical applications.

Definition 3.9 (Bounded-Norm Inner Product over $\mathbb{Z}$)**.** This function family $\mathcal{F}^{\mathsf{IP}}_{m,X,Y}$, where $m, X, Y \in \mathbb{N}$, consists of functions $f_{\mathbf{y}} : \mathcal{X} \to \mathbb{Z}$, where $\mathbf{y} \in \mathbb{Z}^m$ s.t. $||\mathbf{y}||_\infty \leq Y$ and $\mathcal{X} = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^m, ||\mathbf{x}||_\infty \leq X\}$. For all $\mathbf{x} \in \mathcal{X}$ we define the function as

$$f_{\mathbf{y}}(\mathbf{x}) := \langle \mathbf{x}, \mathbf{y} \rangle.$$

Definition 3.10 (Multi-Input Bounded-Norm Inner Product over $\mathbb{Z}$)**.** This family $\mathcal{F}^{\mathsf{MIP}}_{m,\mu,X,Y}$, where $m, \mu, X, Y \in \mathbb{N}$, consists of functions $f_{\mathbf{y}_1,\ldots,\mathbf{y}_\mu} : \mathcal{X}^\mu \to \mathbb{Z}$, where $\mathbf{y}_i \in \mathbb{Z}^m$ s.t. $||\mathbf{y}_i||_\infty \leq Y$ and $\mathcal{X}$ is defined the same as above. For all $(\mathbf{x}_1, \ldots, \mathbf{x}_\mu) \in \mathcal{X}^\mu$, we define the function as

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

Next, we define a function class for unbounded (single-input) inner products, which is related to Chapter 5.

Definition 3.11 (Bounded-Norm Unbounded Inner Product over $\mathbb{Z}$)**.** Here, we define the (E:con, K:sep, D:ct-dom) setting. This function family $\mathcal{F}^{\mathsf{UIP}}_{X,Y}$, where $X, Y \in \mathbb{N}$, consists of functions $f_{S,\mathbf{y}} : \mathcal{X} \to \mathbb{Z} \cup \{\bot\}$, where $S \subset \mathbb{N}$, $\mathbf{y} := (y_i)_{i\in S} \in \mathbb{Z}^S$ s.t. $||\mathbf{y}||_\infty \leq Y$, and $\mathcal{X} = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_{i\in\mathbb{N}} \mathbb{Z}^i, ||\mathbf{x}||_\infty \leq X\}$. We define the function for all $\mathbf{x} \in \mathcal{X}$ as

$$f_{S,\mathbf{y}}(\mathbf{x}) := \begin{cases} \sum_{i\in S} x_i y_i & (S \subseteq [m]) \\ \bot & (S \nsubseteq [m]) \end{cases},$$

where m is the number of elements of $\mathbf{x}$.

The correctness and security of our schemes follow Definitions 3.6 to 3.8 by adapting them to function classes for inner products defined above. We remark one thing on the security game for unbounded inner products. The straightforward adaptation of the restriction Eq. (3.1) to unbounded inner products results in the following restriction. For all $j \in [q_{\mathsf{ct}}]$ and $\ell \in [q_{\mathsf{sk}}]$, if $S_\ell \subseteq [m_j]$, then

$$\sum_{i\in S_\ell} x^0_{j,i} y_{\ell,i} = \sum_{i\in S_\ell} x^1_{j,i} y_{\ell,i} \quad \text{(for public-key FE)}, \tag{3.5}$$

$$\sum_{i\in S_\ell} x^0_{j,i} y^0_{\ell,i} = \sum_{i\in S_\ell} x^1_{j,i} y^1_{\ell,i} \quad \text{(for private-key FE)}. \tag{3.6}$$

However, our unbounded IPFE schemes do not hide the length of an underlying vector. Thus, we additionally assume that $\mathbf{x}^0_j$ and $\mathbf{x}^1_j$ have the same length in the security game for FE schemes in Chapter 5.

Chapter 4

Efficient Function-Hiding Inner Product Functional Encryption

In this chapter, we present our contribution in [TAO20], which proposes a more efficient function-hiding IPFE scheme from a weaker assumption than previous ones.

4.1 Technical Overview

From a technical view point, it is generally difficult to achieve more efficient cryptographic schemes from weaker assumptions. In our case, however, we achieve a more efficient scheme from the weaker assumption than those by Datta *et al.* [DDM16]. We have overcome this difficulty by developing a new security proof strategy. A bit more precisely, we refine the process of the hybrid argument based on DPVS framework. It allows us to prove the security of the scheme in a compact space, and the number of elements in the scheme become small. Datta *et al.* employ more complicated but unnecessary steps in the hybrid argument, which requires a large space to prove the security.

One of the main ingredients of the new strategy is the change of the way to reduce the k-Lin problem to game changes. Consider the SXDH assumption (the case of $k = 1$). Roughly speaking, the SXDH assumption says that it is difficult for efficient adversaries to distinguish whether given two two-dimensional vectors span a whole space or not. We use the assumption to argue that it is difficult to distinguish whether a certain ciphertext or secret key is distributed in an $(m + 1)$-dimensional space or an $(m + 2)$-dimensional space, whereas Datta *et al.* use it to argue that whether one is distributed in an m-dimensional space or a $2m$-dimensional space. Thus, we can reduce m dimensions from their scheme. We save further m dimensions by the refinement of hybrid argument design, but we refrain from going into detail here.

4.2 Efficient Function-Hiding Inner Product Functional Encryption

4.2.1 Construction

In this section, we present our effcient fully function-hiding private-key IPFE scheme (FE for $\mathcal{F}^{\mathsf{IP}}_{m,X,Y}$). Let X_λ and Y_λ be norm-bounds, which are polynomials in λ.

$\mathsf{Setup}(1^\lambda, 1^m)$**:** : The setup algorithm takes a security parameter 1^λ and a vector length 1^m and outputs $(\mathsf{pp}, \mathsf{msk})$, where

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \;\; \mathbf{B} \leftarrow \mathsf{GL}_{2m+2k+1}(\mathbb{Z}_p)$$
$$\mathsf{pp} := \mathbb{G}, \;\; \mathsf{msk} := \begin{pmatrix} \mathbf{b}_1, \ldots, \mathbf{b}_m, \mathbf{b}_{2m+1}, \ldots, \mathbf{b}_{2m+k}, \\ \mathbf{b}^*_1, \ldots, \mathbf{b}^*_m, \mathbf{b}^*_{2m+k+1}, \ldots, \mathbf{b}^*_{2m+2k} \end{pmatrix}.$$

In the above, $\mathbf{b}_i$ and $\mathbf{b}^*_i$ denote the i-th rows of $\mathbf{B}$ and $\mathbf{B}^*$, respectively.

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

$$\mathbf{s} \leftarrow \mathbb{Z}^k_p, \;\; \tilde{\mathbf{x}} := (\mathbf{x}, 0^m, \mathbf{s}, 0^k, 0) \in \mathbb{Z}^{2m+2k+1}_p,$$
$$\mathsf{ct} := [\tilde{\mathbf{x}}\mathbf{B}]_1.$$

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

$$\mathbf{r} \leftarrow \mathbb{Z}^k_p, \;\; \tilde{\mathbf{y}} := (\mathbf{y}, 0^m, 0^k, \mathbf{r}, 0) \in \mathbb{Z}^{2m+2k+1}_p,$$
$$\mathsf{sk} := [\tilde{\mathbf{y}}\mathbf{B}^*]_2.$$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk})$**:** : The decryption algorithm computes the pairing of a secret key and a ciphertext as $z := e(\mathsf{ct}, \mathsf{sk})$. Then it searches for d such that $g^d_T = z$ in the range from $-mXY$ to mXY. If it finds d that satisfies $g^d_T = z$, outputs d. Otherwise, outputs $\perp$.

Correctness. Observe that $d := e(\mathsf{ct}, \mathsf{sk}) = [\tilde{\mathbf{x}}\mathbf{B}\mathbf{B}^{-1}\tilde{\mathbf{y}}^\top]_T = [\langle \mathbf{x}, \mathbf{y} \rangle]_T$. Therefore, d that the decryption algorithm outputs is $\langle \mathbf{x}, \mathbf{y} \rangle$.

4.2.2 Security

Lemmas for the Security Proof. We consider the following problems and use them to prove the security of our scheme.

Definition 4.1 (Problem 1)**.** Problem 1 is to guess a bit $\beta \in \{0,1\}$, given (P, T_β), where

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathbf{B} \leftarrow \mathsf{GL}_{2m+2k+1}(\mathbb{Z}_p), \\
&P := \begin{pmatrix} \mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_{2m+k}]_1, \\ [\mathbf{b}^*_1]_2, \ldots, [\mathbf{b}^*_{2m}]_2, [\mathbf{b}^*_{2m+k+1}]_2, \ldots, [\mathbf{b}^*_{2m+2k}]_2 \end{pmatrix}, \\
&\beta \leftarrow \{0,1\}, \quad \mathbf{s} \leftarrow \mathbb{Z}_p^k, \quad u \leftarrow \mathbb{Z}_p, \\
&T_\beta := [(0^{2m}, \mathbf{s}, 0^k, \beta u)\mathbf{B}]_1.
\end{aligned}$$

Definition 4.2 (Problem 2)**.** Problem 2 is to guess a bit $\beta \in \{0,1\}$, given (P, T_β), where

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathbf{B} \leftarrow \mathsf{GL}_{2m+2k+1}(\mathbb{Z}_p), \\
&P := \begin{pmatrix} \mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_{2m+k}]_1, \\ [\mathbf{b}^*_1]_2, \ldots, [\mathbf{b}^*_{2m}]_2, [\mathbf{b}^*_{2m+k+1}]_2, \ldots, [\mathbf{b}^*_{2m+2k}]_2 \end{pmatrix}, \\
&\beta \leftarrow \{0,1\}, \quad \mathbf{r} \leftarrow \mathbb{Z}_p^k, \quad u \leftarrow \mathbb{Z}_p, \\
&T_\beta := [(0^{2m}, 0^k, \mathbf{r}, \beta u)\mathbf{B}^*]_2.
\end{aligned}$$

For a PPT algorithm $\mathcal{A}$, the advantage for Problem n ($n = 1, 2$) is defined as

$$\mathsf{Adv}^{\mathsf{P}n}_{\mathcal{A}}(\lambda) := \left|\mathsf{Pr}[\mathcal{A}(P, T_0) \to 1] - \mathsf{Pr}[\mathcal{A}(P, T_1) \to 1]\right|.$$

Then the following lemmas hold.

Lemma 4.1. *For any PPT adversary $\mathcal{A}$ for Problem 1, there exists a PPT adversary $\mathcal{B}$ for the k-Lin problem such that $\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{k\text{-}\mathsf{Lin}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.*

Lemma 4.2. *For any PPT adversary $\mathcal{A}$ for Problem 2, there exists a PPT adversary $\mathcal{B}$ for the k-Lin problem such that $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{k\text{-}\mathsf{Lin}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.*

Proof of Lemma 4.1. We show that we can construct a PPT adversary $\mathcal{B}$ for the k-Lin problem from any PPT adversary $\mathcal{A}$ for Problem 1. $\mathcal{B}$ is given an instance of the k-Lin problem with $i = 1$, i.e., $(\mathbb{G}, [\mathbf{A}]_1, [\mathbf{t}_\beta]_1)$. Let $\mathbf{M} := (\mathbf{A}||\mathbf{e}_1)^\top$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^\top \in \mathbb{Z}_p^{k+1}$. Observe that $\mathbf{t}_0$ and $\mathbf{t}_1$ are distributed as $\mathbf{t}_0 = (\mathbf{v}, 0)\mathbf{M}$ and $\mathbf{t}_1 = (\mathbf{v}, u)\mathbf{M}$, respectively, where $\mathbf{v} \leftarrow \mathbb{Z}_p^k, u \leftarrow \mathbb{Z}_p$. For $\mathbf{M}^* \in \mathbb{Z}_p^{(k+1)\times(k+1)}$, we denote the upper submatrix of $\mathbf{M}^*$ with k rows by $\overline{\mathbf{M}^*}$ and the lower submatrix with 1 row by $\underline{\mathbf{M}^*}$. Then $\mathcal{B}$ sets a random basis $\mathbf{B}$ as

$$\begin{aligned}
&\mathbf{W} \leftarrow \mathsf{GL}_{2m+2k+1}(\mathbb{Z}_p), \\
&\mathbf{B} := \begin{pmatrix} \mathbf{I}_{2m} & & \\ & \mathbf{A}^\top & \\ & & \mathbf{I}_k \\ & \mathbf{e}_1^\top & \end{pmatrix} \mathbf{W} \in \mathbb{Z}_p^{2m+2k+1}.
\end{aligned}$$

It is not difficult to see that $\mathbf{B}$ is invertible iff $\mathbf{M}$ is invertible, and $\mathbf{M}$ is invertible with the

overwhelming probability. In this case, we have

$$\mathbf{B}^* = \begin{pmatrix} \mathbf{I}_{2m} & & \\ & \overline{\mathbf{M}^*} & \\ & & \mathbf{I}_k \\ & \underline{\mathbf{M}^*} & \end{pmatrix} \mathbf{W}^* \in \mathbb{Z}_p^{2m+2k+1}.$$

Then, $\mathcal{B}$ can compute P from $[\mathbf{A}]_1$:

$$P := \begin{pmatrix} \mathbb{G}, [\mathbf{b}]_1, \ldots, [\mathbf{b}_{2m+k}]_1, \\ [\mathbf{b}_1^*]_2, \ldots, [\mathbf{b}_{2m}^*]_2, [\mathbf{b}_{2m+k+1}^*]_2, \ldots, [\mathbf{b}_{2m+2k}^*]_2 \end{pmatrix}.$$

Finally, $\mathcal{B}$ sets

$$T_\beta := [(0^{2m}, \mathbf{t}_\beta^\top, 0^k)\mathbf{W}]_1 = [(0^{2m}, \mathbf{v}, 0^k, \beta u)\mathbf{B}]_1.$$

Observe that P and T_β are identically distributed to the instance of Problem 1 unless $\mathbf{A}$ is singular or $u = 0$, which occurs with the probability $2^{-\Omega(\lambda)}$. This is because $\mathbf{A}$ is singular iff at least one of $\{a_i\}_{i\in[k]}$ equals to 0, which occurs with probability $1 - (1 - 1/p)^k \leq k/p$. Note that $k = O(1)$ and $p = 2^{\Omega(\lambda)}$. Hence, the lemma holds. □

Proof of Lemma 4.2. We omit the proof of Lemma 4.2 because it is similar to that of Lemma 4.1.

Security Proof of Our Scheme. The proposed scheme is fully function-hiding under the k-Lin assumption. More precisely, the following theorem holds.

Theorem 4.1. *The proposed Priv-IPVE scheme* Π *is fully function-hiding under the* k*-Lin assumption. For any PPT adversary* $\mathcal{A}$*, there exists a PPT adversary* $\mathcal{B}$ *for the* k*-Lin problem such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{FHS},\Pi}(\lambda) \leq 2q_1\mathsf{Adv}_{\mathcal{B}}^{k\text{-}\mathsf{Lin}}(\lambda) + 4q_2\mathsf{Adv}_{\mathcal{B}}^{k\text{-}\mathsf{Lin}}(\lambda) + 2^{-\Omega(\lambda)},$$

where q_1 *is the number of* $\mathcal{A}$*'s secret key queries, and* q_2 *is the number of* $\mathcal{A}$*'s ciphertext queries.*

Proof of Theorem 4.1. For the proof of Theorem 4.1, we use a hybrid argument over a series of games that differ in the construction of the challenge ciphertexts and secret keys. The game sequence proceeds as Table 4.1. It also shows the structures of ciphertexts and secret keys in the end of each game. In the Game 1 sequence, we generate the term $\mathbf{x}^1$ in the $(m+1)$-th to $2m$-th elements of $\tilde{\mathbf{x}}$ for all ciphertexts. In the Game 2 sequence, for all secret keys, we gradually change $\mathbf{y}^0$ in the 1st to m-th elements of $\tilde{\mathbf{y}}$, to $\mathbf{y}^1$ in the $(m+1)$-th to $2m$-th elements. The remaining games are almost the reverse steps of the Game 1 sequence.

Next, we formally describe each game. We frame a part by a box that was changed from a previous game.

Game	$\tilde{\mathbf{x}}$ in Ciphertexts	$\tilde{\mathbf{y}}$ in Secret Keys
Game 0	$(\mathbf{x}^0, 0^m, \mathbf{s}, 0^k, 0)$	$(\mathbf{y}^0, 0^m, 0^k, \mathbf{r}, 0)$
Game 1-1-1		
$\vdots$	$\vdots$	$\vdots$
Game 1-q_2-3	$(\mathbf{x}^0, \mathbf{x}^1, \mathbf{s}, 0^k, 0)$	$(\mathbf{y}^0, 0^m, 0^k, \mathbf{r}, 0)$
Game 2-1-1		
$\vdots$	$\vdots$	$\vdots$
Game 2-q_1-3	$(\mathbf{x}^0, \mathbf{x}^1, \mathbf{s}, 0^k, 0)$	$(0^m, \mathbf{y}^1, 0^k, \mathbf{r}, 0)$
Game 3	$(\mathbf{x}^1, \mathbf{x}^0, \mathbf{s}, 0^k, 0)$	$(\mathbf{y}^1, 0^m, 0^k, \mathbf{r}, 0)$
Game 4	$(\mathbf{x}^1, 0^m, \mathbf{s}, 0^k, 0)$	$(\mathbf{y}^1, 0^m, 0^k, \mathbf{r}, 0)$

Table 4.1: Game sequence with the structures of ciphertexts and secret keys.

Game 0: This game is the real one where the challenger selects 0 as a random bit. That is, for all $j = 1, \ldots, q_1$ and all $\ell = 1, \ldots, q_2$, $\widetilde{\mathbf{y}}_j$ and $\widetilde{\mathbf{x}}_\ell$ in the replies for the j-th secret key query and the ℓ-th ciphertext query correspond to

$$\widetilde{\mathbf{y}}_j := (\mathbf{y}_j^0, 0^m, 0^k, \mathbf{r}_j, 0),$$
$$\widetilde{\mathbf{x}}_\ell := (\mathbf{x}_\ell^0, 0^m, \mathbf{s}_\ell, 0^k, 0).$$

Game 1-μ-1($\mu = 1, \ldots, q_2$)**:** Game 1-0-3 is defined to be Game 0. This game is the same as Game 1-(μ-1)-3 except that $\widetilde{\mathbf{x}}_\mu$ in the reply for the μ-th ciphertext query corresponds to

$$\widetilde{\mathbf{x}}_\mu := (\mathbf{x}_\mu^0, 0^m, \mathbf{s}_\mu, 0^k, \boxed{u}),$$

where $u \leftarrow \mathbb{Z}_p$.

Game 1-μ-2($\mu = 1, \ldots, q_2$)**:** This game is the same as Game 1-μ-1 except that $\widetilde{\mathbf{x}}_\mu$ in the reply for the μ-th ciphertext query corresponds to

$$\widetilde{\mathbf{x}}_\mu := (\mathbf{x}_\mu^0, \boxed{\mathbf{x}_\mu^1}, \mathbf{s}_\mu, 0^k, u).$$

Game 1-μ-3($\mu = 1, \ldots, q_2$)**:** This game is the same as Game 1-μ-2 except that $\widetilde{\mathbf{x}}_\mu$ in the reply for the μ-th ciphertext query corresponds to

$$\widetilde{\mathbf{x}}_\mu := (\mathbf{x}_\mu^0, \mathbf{x}_\mu^1, \mathbf{s}_\mu, 0^k, \boxed{0}).$$

Game 2-ν-1($\nu = 1, \ldots, q_1$)**:** Game 2-0-3 is defined to be 1-q_2-3. This game is the same as Game 2-(ν-1)-3 except that $\widetilde{\mathbf{y}}_\nu$ in the reply for the ν-th secret key query corresponds to

$$\widetilde{\mathbf{y}}_\nu := (\mathbf{y}_\nu^0, 0^m, 0^k, \mathbf{r}_\nu, \boxed{u}),$$

where $u \leftarrow \mathbb{Z}_p$.

Game 2-ν-2($\nu = 1, \ldots, q_1$)**:** This game is the same as Game 2-ν-1 except that $\widetilde{\mathbf{y}}_\nu$ in the reply for the ν-th secret key query corresponds to

$$\widetilde{\mathbf{y}}_\nu := (\boxed{0^m, \mathbf{y}_\nu^1}, 0^k, \mathbf{r}_\nu, u).$$

Game 2-ν-3($\nu = 1, \ldots, q_1$)**:** This game is the same as Game 2-ν-2 except that $\widetilde{\mathbf{y}}_\nu$ in the reply for the ν-th secret key query corresponds to

$$\widetilde{\mathbf{y}}_\nu := (0^m, \mathbf{y}_\nu^1, 0^k, \mathbf{r}_\nu, \boxed{0}).$$

Game 3: This game is the same as Game 2-q_2-3 except that, for all $j = 1, \ldots, q_1$ and all $\ell = 1, \ldots, q_2$, $\widetilde{\mathbf{y}}_j$ and $\widetilde{\mathbf{x}}_\ell$ in the replies for the j-th secret key query and the ℓ-th ciphertext query correspond to

$$\widetilde{\mathbf{y}}_j := (\boxed{\mathbf{y}_j^1, 0^m}, 0^k, \mathbf{r}_j, 0),$$
$$\widetilde{\mathbf{x}}_\ell := (\boxed{\mathbf{x}_\ell^1, \mathbf{x}_\ell^0}, \mathbf{s}_\ell, 0^k, 0).$$

Game 4: This game is the same as Game 3 except that, for all $\ell = 1, \ldots, q_2$, $\widetilde{\mathbf{x}}_\ell$ in the reply for the ℓ-th ciphertext query corresponds to

$$\widetilde{\mathbf{x}}_\ell := (\mathbf{x}_\ell^1, \boxed{0^m}, \mathbf{s}_\ell, 0^k, 0).$$

Note that this game is the real one where the challenger selects 1 as a random bit.

In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game X as E_X.

Lemma 4.3. *For any PPT distinguisher $\mathcal{A}$ between Game 1-(μ-1)-3 and Game 1-μ-1, there exists a PPT algorithm $\mathcal{B}$ for Problem 1, such that for any security parameter λ,*

$$\left|\mathsf{Pr}[E_{1\text{-}(\mu-1)\text{-}3}] - \mathsf{Pr}[E_{1\text{-}\mu\text{-}1}]\right| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda).$$

Proof. We show that it is possible to construct a PPT algorithm $\mathcal{B}$ for Problem 1 using any PPT distinguisher $\mathcal{A}$ between Game 1-(μ-1)-3 and Game 1-μ-1 as a blackbox. $\mathcal{B}$ takes a role to $\mathcal{A}$ as a challenger of the security game.

1. $\mathcal{B}$ is given a Problem 1 instance (P, T_β).
2. $\mathcal{B}$ gives $\mathbb{G}$ to $\mathcal{A}$ as pp.
3. $\mathcal{B}$ computes sk_j using vectors in P when $\mathcal{A}$ makes a query on $(\mathbf{y}_j^0, \mathbf{y}_j^1)$.
4. $\mathcal{B}$ computes $\mathsf{ct}_\ell (\ell \neq \mu)$ using vectors in P when $\mathcal{A}$ makes a query on $(\mathbf{x}_\ell^0, \mathbf{x}_\ell^1)$.
5. $\mathcal{B}$ computes ct_μ when $\mathcal{A}$ makes a query on $(\mathbf{x}_\mu^0, \mathbf{x}_\mu^1)$ as

$$\mathsf{ct}_\mu := [(\mathbf{x}_\mu^0, 0^{m+2k+1})\mathbf{B}]_1 + T_\beta = [(\mathbf{x}_\mu^0, 0^m, \mathbf{s}_\mu, 0^k, \beta u)\mathbf{B}]_1,$$

where $\mathbf{s}_\mu := \mathbf{s}$. Observe that $[(\mathbf{x}_\mu^0, 0^{m+2k+1})\mathbf{B}]_1$ is computable with the vectors in P.

6. When $\mathcal{A}$ outputs β', $\mathcal{B}$ outputs β' as it is.

If $\beta = 0$, $\mathcal{A}$'s view is the same as that in Game 1-(μ-1)-3, and if $\beta = 1$, $\mathcal{B}$'s view is the same as that in Game 1-μ-1. □

Lemma 4.4. *For any PPT distinguisher $\mathcal{A}$ between Game 1-μ-1 and Game 1-μ-2,*

$$\mathsf{Pr}[E_{1\text{-}\mu\text{-}1}] = \mathsf{Pr}[E_{1\text{-}\mu\text{-}2}].$$

Proof. We demonstrate that $\mathcal{A}$'s view in Game 1-μ-1 is the same as that in Game 1-μ-2. For the purpose, we define a new basis and its dual $(\mathbf{D}, \mathbf{D}^*)$ as

$$\mathbf{D} := \left(\begin{array}{cc:c} \multicolumn{2}{c:}{\multirow{2}{*}{$\mathbf{I}_{2m+2k}$}} & \\ & & \\ \hdashline 0^m & -\frac{\mathbf{x}^1_\mu}{u} \quad 0^{2k} & 1 \end{array}\right)\mathbf{B}, \quad \mathbf{D}^* := \left(\begin{array}{c:c} \multirow{2}{*}{$\mathbf{I}_{2m+2k}$} & 0^m \\ & \frac{\mathbf{x}^{1\top}_\mu}{u} \\ & 0^{2k} \\ \hdashline & 1 \end{array}\right)\mathbf{B}^*.$$

Observe that $(\mathbf{D}, \mathbf{D}^*)$ are random dual orthonormal bases because $(\mathbf{B}, \mathbf{B}^*)$ are random dual orthonormal bases. Then, the coefficients of the secret keys and ciphertexts except the μ-th one are not changed;

$$\begin{aligned}
\mathsf{sk}_j =& [(\mathbf{y}^0_j, 0^m, 0^k, \mathbf{r}_j, 0)\mathbf{B}^*]_2 = [(\mathbf{y}^0_j, 0^m, 0^k, \mathbf{r}_j, 0)\mathbf{D}^*]_2, \\
\mathsf{ct}_\ell =& \begin{cases} [(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, 0)\mathbf{B}]_1 = [(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, 0)\mathbf{D}]_1 & (\ell < \mu) \\ [(\mathbf{x}^0_\ell, 0^m, \mathbf{s}_\ell, 0^k, 0)\mathbf{B}]_1 = [(\mathbf{x}^0_\ell, 0^m, \mathbf{s}_\ell, 0^k, 0)\mathbf{D}]_1 & (\ell > \mu) \end{cases}.
\end{aligned}$$

On the other hand, the μ-th ciphertext is written as

$$\begin{aligned}
\mathsf{ct}_\mu &= [(\mathbf{x}^0_\mu, 0^m, \mathbf{s}_\mu, 0^k, u)\mathbf{B}]_1 \\
&= \left[(\mathbf{x}^0_\mu, 0^m, \mathbf{s}_\mu, 0^k, u) \left(\begin{array}{cc:c} \multicolumn{2}{c:}{\multirow{2}{*}{$\mathbf{I}_{2m+2k}$}} & \\ & & \\ \hdashline 0^m & \frac{\mathbf{x}^1_\mu}{u} \quad 0^{2k} & 1 \end{array}\right)\mathbf{D}\right]_1 \\
&= [(\mathbf{x}^0_\mu, \mathbf{x}^1_\mu, \mathbf{s}_\mu, 0^k, u)\mathbf{D}]_1.
\end{aligned}$$

Therefore, $\mathcal{A}$'s views in both games are information-theoretically identical. □

Lemma 4.5. *For any PPT distinguisher $\mathcal{A}$ between Game 1-μ-2 and Game 1-μ-3, there exists a PPT algorithm $\mathcal{B}$ for Problem 1, such that for any security parameter λ,*

$$|\mathsf{Pr}[E_{1\text{-}\mu\text{-}2}] - \mathsf{Pr}[E_{1\text{-}\mu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda).$$

Proof. The proof of Lemma 4.5 is similar to that of Lemma 4.3, so we omit the proof. □

Lemma 4.6. *For any PPT distinguisher $\mathcal{A}$ between Game 2-(ν-1)-3 and Game 2-ν-1, there exists a PPT algorithm $\mathcal{B}$ for Problem 2, such that for any security parameter λ,*

$$\left|\mathsf{Pr}[E_{2\text{-}(\nu-1)\text{-}3}] - \mathsf{Pr}[E_{2\text{-}\nu\text{-}1}]\right| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda).$$

Proof. We demonstrate that it is possible to construct a PPT algorithm $\mathcal{B}$ for Problem 2 using any PPT distinguisher $\mathcal{A}$ between Game 2-(ν-1)-3 and Game 2-ν-1 as a blackbox. $\mathcal{B}$ takes a role to $\mathcal{A}$ as a challenger of the security game.

1. $\mathcal{B}$ is given a Problem 2 instance (P, T_β).
2. $\mathcal{B}$ gives $\mathbb{G}$ to $\mathcal{A}$ as pp.
3. $\mathcal{B}$ computes ct_ℓ using vectors in P when $\mathcal{A}$ makes a query on $(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell)$.
4. $\mathcal{B}$ computes $\mathsf{sk}_j (j \neq \nu)$ using vectors in P when $\mathcal{A}$ makes a query on $(\mathbf{y}^0_j, \mathbf{y}^1_j)$.
5. $\mathcal{B}$ computes sk_ν when $\mathcal{A}$ makes a query on $(\mathbf{y}^0_\nu, \mathbf{y}^1_\nu)$ as
$$\mathsf{sk}_\nu := [(\mathbf{y}^0_\nu, 0^{m+2k+1})\mathbf{B}^*]_2 + T_\beta = [(\mathbf{y}^0_\nu, 0^m, 0^k, \mathbf{r}_\nu, \beta u)\mathbf{B}^*]_2,$$
where $\mathbf{r}_\nu := \mathbf{r}$. Observe that $[(\mathbf{y}^0_\nu, 0^{m+2k+1})\mathbf{B}^*]_2$ is computable with the vectors in P.
6. When $\mathcal{A}$ outputs β', $\mathcal{B}$ outputs β' as it is.

If $\beta = 0$, $\mathcal{A}$'s view is the same as that in Game 2-(ν-1)-3, and if $\beta = 1$, $\mathcal{A}$'s view is the same as that in Game 2-ν-1. □

Lemma 4.7. *For any PPT distinguisher $\mathcal{A}$ between Game 2-ν-1 and Game 2-ν-2,*
$$\mathsf{Pr}[E_{2\text{-}\nu\text{-}1}] = \mathsf{Pr}[E_{2\text{-}\nu\text{-}2}].$$

Proof. We demonstrate that $\mathcal{A}$'s view in Game 2-ν-1 is the same as that in Game 2-ν-2. For that purpose, we define new basis and its dual $(\mathbf{D}, \mathbf{D}^*)$ as
$$\mathbf{D} := \left(\begin{array}{c:c} \mathbf{I}_{2m+2k} & \begin{array}{c} -\frac{\mathbf{y}^{0\top}_\nu}{u} \\ \frac{\mathbf{y}^{1\top}_\nu}{u} \\ 0^{2k} \end{array} \\ \hdashline & 1 \end{array}\right)\mathbf{B}, \quad \mathbf{D}^* := \left(\begin{array}{c:c} \mathbf{I}_{2m+2k} & \\ \hdashline \frac{\mathbf{y}^0_\nu}{u} \quad -\frac{\mathbf{y}^1_\nu}{u} \quad 0^{2k} & 1 \end{array}\right)\mathbf{B}^*.$$
Observe that $(\mathbf{D}, \mathbf{D}^*)$ are random dual orthonormal bases because $(\mathbf{B}, \mathbf{B}^*)$ are random dual orthonormal bases. Then, the coefficients of the secret keys except the ν-th one are not changed;
$$\mathsf{sk}_j = \begin{cases} [(0^m, \mathbf{y}^1_j, 0^k, \mathbf{r}_j, 0)\mathbf{B}^*]_2 = [(0^m, \mathbf{y}^1_j, 0^k, \mathbf{r}_j, 0)\mathbf{D}^*]_2 & (j < \nu) \\ [(\mathbf{y}^0_j, 0^m, 0^k, \mathbf{r}_j, 0)\mathbf{B}^*]_2 = [(\mathbf{y}^0_j, 0^m, 0^k, \mathbf{r}_j, 0)\mathbf{D}^*]_2 & (j > \nu) \end{cases}.$$
Next, we observe the ν-th secret key;
$$\begin{aligned} \mathsf{sk}_\nu =& [(\mathbf{y}^0_\nu, 0^m, 0^k, \mathbf{r}_\nu, u)\mathbf{B}^*]_2 \\ =& \left[(\mathbf{y}^0_\nu, 0^m, 0^k, \mathbf{r}_\nu, u)\left(\begin{array}{c:c} \mathbf{I}_{2m+2k} & \\ \hdashline -\frac{\mathbf{y}^0_\nu}{u} \quad \frac{\mathbf{y}^1_\nu}{u} \quad 0^{2k} & 1 \end{array}\right)\mathbf{D}^*\right]_2 \\ =& [(0^m, \mathbf{y}^1_\nu, 0^k, \mathbf{r}_\nu, u)\mathbf{D}^*]_2. \end{aligned}$$

Lastly, we check that the coefficients of ciphertexts are not changed;

$$\begin{aligned}
\mathsf{ct}_\ell =& [(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, 0)\mathbf{B}]_1 \\
=& \left[(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, 0) \left(\begin{array}{c:c} \mathbf{I}_{2m+2k} & \begin{array}{c} \frac{\mathbf{y}^{0\top}_\nu}{u} \\ -\frac{\mathbf{y}^{1\top}_\nu}{u} \\ 0^{2k} \end{array} \\ \hdashline & 1 \end{array} \right) \mathbf{D} \right]_1 \\
=& \left[\left(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, \frac{\mathbf{x}^0_\ell \cdot \mathbf{y}^0_\nu - \mathbf{x}^1_\ell \cdot \mathbf{y}^1_\nu}{u} \right) \mathbf{D} \right]_1 \\
=& [(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, 0)\mathbf{D}]_1.
\end{aligned}$$

In the fifth line, we use the query restriction on $\mathcal{A}$, i.e., $\mathbf{x}^0_\ell \cdot \mathbf{y}^0_\nu = \mathbf{x}^1_\ell \cdot \mathbf{y}^1_\nu$. Therefore, $\mathcal{A}$'s views in both games are information-theoretically identical. □

Lemma 4.8. *For any PPT distinguisher $\mathcal{A}$ between Game 2-ν-2 and Game 2-ν-3, there exists a PPT algorithm $\mathcal{B}$ for Problem 2, such that for any security parameter λ,*

$$|\mathsf{Pr}[E_{2\text{-}\nu\text{-}2}] - \mathsf{Pr}[E_{2\text{-}\nu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda).$$

Proof. The proof of Lemma 4.8 is similar to that of Lemma 4.6, so we omit the proof. □

Lemma 4.9. *For any PPT distinguisher $\mathcal{A}$ between Game 2-q_1-3 and Game 3,*

$$\mathsf{Pr}[E_{2\text{-}q_1\text{-}3}] = \mathsf{Pr}[E_3].$$

Proof. We demonstrate that $\mathcal{A}$'s view in Game 2-q_1-3 is the same as that in Game 3. For that purpose, we define new basis and its dual $(\mathbf{D}, \mathbf{D}^*)$ as

$$\mathbf{D} := \begin{pmatrix} & \mathbf{I}_m & \\ \mathbf{I}_m & & \\ & & \mathbf{I}_{2k+1} \end{pmatrix} \mathbf{B}, \quad \mathbf{D}^* := \begin{pmatrix} & \mathbf{I}_m & \\ \mathbf{I}_m & & \\ & & \mathbf{I}_{2k+1} \end{pmatrix} \mathbf{B}^*.$$

Observe that $(\mathbf{D}, \mathbf{D}^*)$ are random dual orthonormal bases because $(\mathbf{B}, \mathbf{B}^*)$ are random dual orthonormal bases. These bases are dual orthonormal. Then we can see that

$$\begin{aligned}
\mathsf{sk}_j &= [(0^m, \mathbf{y}^1_j, 0^k, \mathbf{r}_j, 0)\mathbf{B}^*]_2 = [(\mathbf{y}^1_j, 0^m, 0^k, \mathbf{r}_j, 0)\mathbf{D}^*]_2, \\
\mathsf{ct}_\ell &= [(\mathbf{x}^0_\ell, \mathbf{x}^1_\ell, \mathbf{s}_\ell, 0^k, 0)\mathbf{B}]_1 = [(\mathbf{x}^1_\ell, \mathbf{x}^0_\ell, \mathbf{s}_\ell, 0^k, 0)\mathbf{D}]_1.
\end{aligned}$$

Thus, $\mathcal{A}$'s views in both games are identical. □

Lemma 4.10. *For any PPT distinguisher $\mathcal{A}$ between Game 3 and Game 4, there exists a PPT algorithm $\mathcal{B}$ for Problem 1, such that for any security parameter λ,*

$$|\mathsf{Pr}[E_3] - \mathsf{Pr}[E_4]| \leq 2q_2\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda).$$

Proof. The difference between Game 3 and Game 4 is almost the same as that between Game 0 and Game 1-q_2-3, just switching $\mathbf{x}^0_\ell$ with $\mathbf{x}^1_\ell$, and $\mathbf{y}^0_j$ with $\mathbf{y}^1_j$. □

Thanks to Lemma 4.1 to Lemma 4.10, Theorem 4.1 holds. □

4.3 Conclusion of Chapter 4

In Chapter 4, we constructed a function-hiding IPFE scheme that is more efficient than previous ones. The function-hiding property is essential when not only an objective data but also a delegated computation is sensitive. Prior to our work, there were two function-hiding IPFE schemes: one satisfies only a weaker security definition [BJK15] and the other is inefficient [DDM16]. We refined the previous proof techniques and proved the fully function-hiding property in the more efficient scheme. Additionally, we use the weaker k-Lin assumption instead of the SXDH assumption, which is used in the previous works [BJK15, DDM16].

Chapter 5

Unbounded Inner Product Functional Encryption

In this chapter, we present our contribution in [TT20], which defines unbounded inner product functional encryption and constructs first concrete schemes in both private-key and public-key settings.

5.1 Technical Overview

5.1.1 Private-key Unbounded Inner Product Functional Encryption

Our starting point is the fully function hiding unbounded multi-input IPFE (MIPFE) scheme proposed by Datta *et al.* [DOT18]. In an unbounded MIPFE scheme, an index space for slots are not determined in the setup phase. Then, roughly speaking, an encryption algorithm can generate a ciphertext that corresponds to a vector $\mathbf{x}$ and an arbitrary index $i \in \mathbb{N}$. Also, a key generation algorithm can issue a secret key that is associated with indexed vectors $(S, \{\mathbf{y}_i\}_{i \in S})$ for an arbitrary set $S \subset \mathbb{N}$. Only if a decryptor has all ciphertexts corresponding to elements of the set S, i.e., $\{\mathsf{ct}_i := \mathsf{MIPFE.Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x}_i)\}_{i \in S}$, the secret key for S can be used for legitimate decryption and reveals $\sum_{i \in S} \langle \mathbf{x}_i, \mathbf{y}_i \rangle$. Their scheme is based on the dual pairing vector spaces (DPVS) framework introduced by Okamoto and Takashima [OT10] and utilizes a pseudorandom function (PRF) to handle an unbounded index space. Consider the unbounded MIPFE scheme in which the vector length is set to 1 and observe that such a scheme already serves the function of UIPFE in the D:ct-dom setting. More precisely, to encrypt $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$, the encryption algorithm computes $\mathsf{ct}_i := \mathsf{MIPFE.Enc}(\mathsf{pp}, \mathsf{msk}, i, x_i)$ for all $i \in [m]$ and set $\mathsf{ct} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_m)$. In key generation for an indexed vector $(S, \mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S)$, the key generation algorithm computes $\mathsf{sk} := \mathsf{MIPFE.KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y})$. Then $\mathsf{MIPFE.Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk})$ outputs $\sum_{i \in S} x_i y_i$. However, this construction allows recomposition of ciphertexts due to the property of MIPFE. That is, for $\mathsf{ct}_1 := (\mathsf{ct}_{1,1}, \ldots, \mathsf{ct}_{1,m})$ and $\mathsf{ct}_2 := (\mathsf{ct}_{2,1}, \ldots, \mathsf{ct}_{2,m})$, we can decrypt a ciphertext like $(\mathsf{ct}_{1,1}, \mathsf{ct}_{2,2}, \ldots, \mathsf{ct}_{2,m})$ correctly whereas UIPFE should not allow such recomposition of ciphertexts.

To prevent such recomposition, each ciphertext of our scheme has a unique randomness that all elements in a ciphertext share. Decryption is possible only if an input ciphertext has a consistent randomness, so this unique randomness prevents recomposed ciphertexts from being decrypted correctly. Essentially, a ciphertext for index i of the MIPFE scheme by Datta *et al.* has a form like $[\mathbf{c}_i]_1 := [(x_i, 1)\mathbf{B}_i]_1$ and each element of a secret key has a form like $[\mathbf{k}_i]_2 := [(y_i, r_i)\mathbf{B}_i^*]_2$, where $\mathbf{B}_i$ is a 2×2 regular matrix, $\mathbf{B}_i^* := (\mathbf{B}_i^{-1})^\top$, and r_i are random elements in $\mathbb{Z}_p$ s.t. $\sum_{i \in S} r_i = 0$. Bases $\mathbf{B}_i$ are generated unboundedly with a PRF. A decryption algorithm computes $[\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle]_T$ and it reveals the inner product $\sum_{i \in S}(x_i y_i + r_i) = \sum_{i \in S} x_i y_i$. In this construction, switching elements of one ciphertext that have the same indices as others does not affect the correct decryption. On the other hand, an element of one ciphertext corresponding to index i of our scheme has a form like $[\mathbf{c}_i]_1 := [(x_i, z)\mathbf{B}_i]_1$ where z is a unique randomness for each ciphertext, whereas each element of a secret key is the same as in the MIPFE scheme. Then it is easy to confirm that unless all $\mathbf{c}_i$ for $i \in S$ have the same randomness, $[\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle]_T$ does not reveal the inner product $\sum_{i \in S} x_i y_i$ and this construction prevents recomposition of ciphertexts.

Although the concept of the construction is simple, the security proof of the scheme is rather complicated. The basic proof strategy of our scheme is the same as that by Tomida *et al.* [TAO16], who proposed a fully function hiding *bounded* IPFE scheme, and this strategy is also employed in [DOT18]. In the case of unbounded MIPFE and UIPFE, however, we encounter a new challenging problem that does not appear in *bounded* IPFE: how to prove collusion resistance against illegitimate secret keys queried by an adversary. More precisely, in the D:ct-dom setting, secret keys whose index sets are not included in the index set of a ciphertext must be useless to decrypt the ciphertext even if their owners collude. For example, an owner of a ciphertext ct_1 for a index set {1,2,3} and two secret keys sk_1 and sk_2 for index sets {1,2,4} and {3,4} respectively must not learn any information about underlying vectors in the ciphertext and secret keys.

In the context of unbounded MIPFE, the problem was solved by cleverly utilizing symmetric key encryption (SKE). Briefly, ciphertexts for index i contain a secret key of SKE that is unique to the index i. On the other hand, a secret key of unbounded MIPFE for an index set S is iteratively encrypted by SKE with all secret keys of SKE in the set S. Then, unless an owner of the secret key for a set S has all ciphertexts in the set S, he or she cannot decrypt the secret key of unbounded MIPFE encrypted by SKE and the secret key is useless to derive information from ciphertexts corresponding to any proper subset of S. Due to UIPFE not allowing the recomposition of ciphertexts, however, we cannot apply a similar technique to UIPFE schemes.

To solve this problem, we introduce a new proof strategy. In fully function hiding scheme, we consider an adversary that can query many ciphertexts and secret keys. First, we generate a situation where it is sufficient to consider only one ciphertext and all secret keys by using hidden spaces of DPVS framework. We can consider that this is a kind of dual system methodology by Waters [Wat09], which allows us to reduce the problem of a security for many keys to that for one key [Wee14]. Then what we need to do next is to ensure that illegitimate keys are useless to decrypt the ciphertext. For the purpose, we randomize all elements in illegitimate secret keys whose indices are out of the index set of the ciphertext by computational argument. That is, the randomization is indistinguishable for all probabilistic polynomial time (PPT) adversaries under

the SXDH assumption. In the above simple example, it means that the elements for index 4 in both secret keys are randomized. The intuitive reason to take this step is to ensure that partial decryption does not leak any information on underlying vectors. That is, in the above example, one can correctly compute the term $x_i y_i$ for indices 1 and 2 with sk_1 and 3 with sk_2, which is masked by the term zr_i. What we want to prove here is that the all zr_i terms are indistinguishable from independently random elements in $\mathbb{Z}_p$ and they completely hide the terms $x_i y_i$. Recall that elements in each secret key contain the random numbers r_i such that $\sum_{i \in S} r_i = 0$. Then, if at least one of r_is in each secret key is randomized, entire r_is become completely random elements in $\mathbb{Z}_p$. At this point, partial decryption with illegitimate secret keys reveals no meaningful information and we can complete the proof.

5.1.2 Public-key Unbounded Inner Product Functional Encryption

Our public-key UIPFE scheme is technically more intricate than our private-key one. Because we do not need to publish any information for encryption in the private-key UIPFE scheme, we can utilize PRFs to generate dual orthonormal bases unboundedly, which is necessary for encryption. More precisely, an encryption algorithm generates a basis for index i as $F_K(i)$ where F_K is a PRF, and encode the i-th element of the vector using the basis. In the public-key setting, however, a setup algorithm needs to publish information that is needed to encrypt vectors. Thus an encryptor cannot utilize PRFs to generate bases because if a key of a PRF is public, the output is no longer pseudorandom.

Our approach to overcome this problem is an indexing technique [OT12b], which is introduced to construct unbounded inner product encryption and attribute based encryption (ABE) schemes. Briefly, we add a two-dimensional prefix that specifies an index to a vector to be encoded, and only if the indices of a ciphertext and a secret key are equal, the correct inner product value is computable. In a ciphertext side, an encoding of the i-th element of a vector $\mathbf{x} := (x_1, \ldots, x_m)$ is the form like $[\mathbf{c}_i]_1 := [(\pi_i(1, i), x_i, z)\mathbf{B}]_1$ and in a secret key side, the index j of an indexed vector $(S, \mathbf{y} := (y_j)_{j \in S})$ is encoded as $[\mathbf{k}_j]_2 := [(\rho_j(-j, 1), y_j, r_j)\mathbf{B}^*]_2$. Then, although all indices share the same dual orthonormal bases, $[\langle \mathbf{c}_i, \mathbf{k}_j \rangle]_T$ reveals the meaningful value only if $i = j$. By this construction, each element in ciphertexts and secret keys is encoded as if dual orthonormal bases that are unique to each index were used.

The basic concept of the security proof of our public-key scheme is also similar to that in [OT12b]. That is, we prove lemmas that say that normal ciphertexts and secret keys are indistinguishable from ones encoded on "somewhat" random dual orthonormal bases for each index by amplifying the entropy of the two-dimensional prefix. More concretely, we use a kind of the following relation in the security proof. Note that it is just a toy example for an intuitive explanation and an informal one. That is, for any polynomial $m := m(\lambda)$, we have the computational indistinguishability:

$$\left\{ \begin{array}{l} [(\pi_i(1, i), x_i, z, \ldots)\mathbf{B}]_1 \\ [(\rho_i(-i, 1), y_i, r_i, \ldots)\mathbf{B}^*]_2 \end{array} \right\}_{i \in [m]} \approx_c \left\{ \begin{array}{l} [(\pi_i(1, i), x_i, z, \ldots)\mathbf{D}_i]_1 \\ [(\rho_i(-i, 1), y_i, r_i, \ldots)\mathbf{D}_i^*]_2 \end{array} \right\}_{i \in [m]},$$

where $\{\pi_i\}_{i\in[m]}, \{\rho_i\}_{i\in[m]} \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ and $\mathbf{D}_i := \mathbf{W}_i\mathbf{B}$. LHS represents normal elements of a ciphertext and secret key, and RHS represents elements of ones encoded on "somewhat" random dual orthonormal bases for each index. Here, each $\mathbf{W}_i$ need not be a completely random matrix, and it is sufficient if $\mathbf{W}_i$ is chosen from some specific distribution for our security proof. This is why we call $\mathbf{D}_i$ "somewhat" random. At this point, we can use the proof strategy similar to that of the private-key IPFE scheme because dual orthonormal bases are generated somewhat randomly for each index and we have a similar situation to the private-key IPFE scheme. Although the top-level concept of the techniques are similar to [OT12b], i.e., indexing and entropy amplification, we cannot directly use their techniques because the security proof of our scheme is completely different from that of their scheme. Therefore, we managed to tailor lemmas of entropy amplification suitable for our scheme.

5.1.3 Discussion

In this thesis, we cannot achieve the schemes that have the following two features. We quickly discuss the difficulty about them.

Public-key UIPFE schemes without pairing. We briefly explain why constructing unbounded public-key IPFE schemes without pairing is difficult. Differently from bounded IPFE schemes, unbounded IPFE schemes have a kind of functionality that ABE schemes support. That is, a ciphertext and secret key are associated with sets U and S, respectively, and decryption is possible only when U and S satisfy some inclusive relation, e.g., $U \supseteq S$. Because we know that ABE schemes based on cyclic groups need pairings, unbounded IPFE schemes seem to require pairings similarly.

Adaptively secure (E:sep, K:sep) UIPFE schemes. As we mentioned, our proof strategy needs to guess an index set of a ciphertext and inherently we cannot apply it to (E:sep, K:sep) schemes with adaptive security. We consider that this difficulty is similar to that to prove adaptive security of multi-use KP-ABE from static assumptions (this problem is solved in the semi-adaptive setting [CW14]). That is, the reduction algorithm needs to embed the instance of an underlying problem into secret keys depending on the instance that the adversary outputs in the challenge phase. Hence, the difficulty disappears in the semi-adaptive setting because the reduction knows the challenge instance before it simulates secret keys. We know that we can obtain adaptively secure multi-use KP-ABE from so-called q-type assumptions [Att14, Att16], then we might be able to obtain adaptively secure (E:sep, K:sep) schemes from q-type assumptions similarly.

5.2 Private-Key Unbounded Inner Product Functional Encryption

We present our main private-key unbounded IPFE scheme, that is, a FE scheme for $\mathcal{F}^{\mathsf{UIP}}_{X,Y}$. Our schemes are based on the DPVS framework (Definition 3.5) introduced by Okamoto and Takashima

[OT10]. We note that our scheme requires that vector lengths are bounded by some polynomials, denoted by $m(\lambda)$ and $s(\lambda)$ in the construction, but they do not have to be fixed at the setup phase. This also goes for our public-key scheme (Section 5.3).

5.2.1 Construction

In the following scheme, norm limits X_λ, Y_λ are some polynomials. Let $\mathcal{F} := \{F_K\}_{K \in \mathcal{K}_\lambda}$ be a PRF family with a key space $\mathcal{K}_\lambda$ consisting of functions $F_K : \{0,1\}^\lambda \to \mathsf{M}_4(\mathbb{Z}_p)$.

$\mathsf{Setup}(1^\lambda)$**:** Takes a security parameter 1^λ and chooses bilinear groups $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$ and a PRF key $K \leftarrow \mathcal{K}_\lambda$. Outputs

$$\mathsf{pp} := \mathbb{G}, \quad \mathsf{msk} := K.$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$**:** Takes $\mathsf{pp}, \mathsf{msk}$ and $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ where $m := m(\lambda)$ is any polynomial. Sets $\mathbf{B}_i := F_K(i)$ and $\mathbf{c}_i := (x_i, 0, z, 0)\mathbf{B}_i \in \mathbb{Z}_p^4$ for all $i \in [m]$, where $z \leftarrow \mathbb{Z}_p$. Outputs

$$\mathsf{ct}_m := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_m]_1).$$

If there exists $i \in [m]$ such that $\mathbf{B}_i$ is a singular matrix, outputs $\perp$.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y})$**:** Takes pp, msk, a non-empty index set $S \subseteq [s]$ where $s := s(\lambda)$ is any polynomial, and an indexed vector $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$. Chooses $\{r_i\}_{i \in S} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i \in S} r_i = 0$. Sets $\mathbf{B}_i := F_K(i)$ and $\mathbf{k}_i := (y_i, 0, r_i, 0)\mathbf{B}_i^* \in \mathbb{Z}_p^4$ for all $i \in S$. Outputs

$$\mathsf{sk}_S := (S, \{[\mathbf{k}_i]_2\}_{i \in S}).$$

If there exists $i \in S$ such that $\mathbf{B}_i$ is a singular matrix, outputs $\perp$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_m, \mathsf{sk}_S)$**:** Takes pp, a ciphertext ct_m for m dimensional vector, and a secret key sk_S for a index set S. If $S \subseteq [m]$, then computes

$$h := \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2),$$

and searches for d s.t. $e(g_1, g_2)^d = h$ exhaustively in the range of $-|S|X_\lambda Y_\lambda$ to $|S|X_\lambda Y_\lambda$. If such d is found, outputs d. Otherwise, outputs $\perp$.

Correctness. Our Priv-UIPFE scheme is correct if $\mathcal{F}$ is a PRF family. We consider the case where for a natural number $m \in \mathbb{N}$, $\mathbf{B}_i := F_K(i)$ for all $i \in [m]$ is invertible. Then, we observe that if $S \subseteq [m]$,

$$h = \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2) = e(g_1, g_2)^{\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle} = e(g_1, g_2)^{\sum_{i \in S} (x_i y_i + z r_i)}.$$

Here we have $\sum_{i \in S} r_i = 0$, then $h = e(g_1, g_2)^{\sum_{i \in S} x_i y_i}$. If $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, $|\sum_{i \in S} x_i y_i| \leq |S| X_\lambda Y_\lambda$ and Dec outputs $\sum_{i \in S} x_i y_i$. Hence, if $\mathbf{B}_i$ for all $i \in [m]$ is invertible without a negligible probability, our scheme is correct. Let $m := m(\lambda)$ be any polynomial. For $i \in [m]$, we have $\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | \mathbf{B}_i \leftarrow \mathsf{M}_4(\mathbb{Z}_p)] = 2^{-\Omega(\lambda)}$ from Lemma 3.1 and $|\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | \mathbf{B}_i \leftarrow \mathsf{M}_4(\mathbb{Z}_p)] - \mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | K \leftarrow \mathcal{K}_\lambda, \mathbf{B}_i := F_K(i)]| \leq \mathsf{negl}(\lambda)$ from the definition of PRF. Consequently, $\mathsf{Pr}[\exists i, \det \mathbf{B}_i = 0 | K \leftarrow \mathcal{K}_\lambda, \mathbf{B}_i := F_K(i)] \leq \mathsf{negl}(\lambda)$.

5.2.2 Security

Theorem 5.1. *Assume that the SXDH assumption holds and $\mathcal{F}$ is a PRF family, then our Priv-UIPFE is fully function-hiding. More formally, let m_{max} be the maximum length of vectors with which $\mathcal{A}$ makes a query to the encryption oracle, then for any PPT adversary $\mathcal{A}$ and security parameter λ, there exists a PPT adversary $\mathcal{B}_1$ for the SXDH and $\mathcal{B}_2$ for the PRF family, we have*

$$\mathsf{Adv}^{\mathsf{Priv\text{-}UIPFE}}_{\mathcal{A}}(\lambda) \leq \{4q_{\mathsf{sk}} + 2(m_{\mathsf{max}}+1)q_{\mathsf{ct}} + 2\}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}_1}(\lambda) + 2\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof outline. The top-level strategy of the proof is similar to that of the proof by Tomida *et al.* [TAO16], although the order of changing the forms of ciphertexts and secret keys is the opposite. In the security proof, we employ a usual hybrid argument and gradually change the forms of ciphertexts and secret keys queried by an adversary from the case of $\beta = 0$ to $\beta = 1$ defined in Definition 3.7. We use the spaces not used in the actual function, i.e., the second and fourth spaces, for the security proof. Intuitively, the second space is a kind of a working space to handle intermediate states between $\beta = 0$ and $\beta = 1$, and the fourth space is utilized to make a situation where we can focus on only one query even if an adversary makes multiple queries. In other words, we can see the fourth space as a semi-functional space of dual system methodology proposed by Waters [Wat09]. First, the form of secret keys is changed from $[(y^0_{\ell,i}, 0, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ to $[(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ in the Game 1 sequence. Next, we change the form of ciphertexts from $[(x^0_{j,i}, 0, z_j, 0)\mathbf{B}_i]_1$ to $[(0, x^1_{j,i}, z_j, 0)\mathbf{B}_i]_1$ in the Game 3 sequence, and here we leverage the game condition Eq. (3.6). Then we switch the first space with the second space as $[(x^1_{j,i}, 0, z_j, 0)\mathbf{B}_i]_1$ and $[(y^1_{\ell,i}, y^0_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$. Finally, the form of secret keys is changed from $[(y^1_{\ell,i}, y^0_{\ell,i}, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ to $[(y^1_{\ell,i}, 0, r_{\ell,i}, 0)\mathbf{B}^*_i]_2$ as the reverse of the Game 1 sequence. The most complicated and important part is the Game 3 sequence, in which we need to deal with the ciphertexts and secret keys that do not satisfy the condition Eq. (3.6). In Game 3 sequence, we change the ciphertexts from 0-side to 1-side one by one, and in the ν-th iteration of Game 3 sequence, we change the ν-th ciphertexts from 0-side to 1-side. For the ν-th ciphertext, we can classify secret keys queried by an adversary into three types. Let m_ν be the length of the ciphertext.

1. The index set S of the secret key is included in $[m_\nu]$, i,e., $\max S \leq m_\nu$.
2. A part of the index set S is included in $[m_\nu]$, i.e., $(\max S > m_\nu) \wedge (\min S \leq m_\nu)$.
3. The index set S and $[m_\nu]$ are disjoint, i.e., $\min S > m_\nu$.

The cumbersome secret keys are type 2 keys because they can correctly decrypt a part of the ciphertext even though they may not satisfy the condition Eq. (3.6). We want to change the form of the ν-th ciphertext from 0-side to 1-side by information-theoretical change in Game 3-ν-1-4, but it does not work without any treatment due to the above property of type 2 keys. Therefore, we manage to randomize type 2 keys from Game 3-ν-1-1 to Game 3-ν-1-3.

Proof. We prove Theorem 5.1 by a series of games. For each game transition, we prove that the difference between probabilities that the adversary $\mathcal{A}$ outputs 1 in both games is negligible.

Game 0: This game is the same as the real security game when $\beta = 0$ in Definition 3.7. That is, the j-th ciphertext query with a pair of vectors $(\mathbf{x}_j^0, \mathbf{x}_j^1) \in (\mathbb{Z}^{m_j})^2$ is replied as

$$\mathbf{c}_{j,i} := (x_{j,i}^0, 0, z_j, 0)\mathbf{B}_i \text{ for all } i \in [m_j]$$
$$\mathsf{ct}_{j,m_j} := ([\mathbf{c}_{j,1}]_1, \ldots, [\mathbf{c}_{j,m_j}]_1).$$

The ℓ-th secret key query with an index set S_ℓ and a pair of vectors $(\mathbf{y}_\ell^0, \mathbf{y}_\ell^1) \in (\mathbb{Z}^{S_\ell})^2$ is replied as

$$\mathbf{k}_{\ell,i} := (y_{\ell,i}^0, 0, r_{\ell,i}, 0)\mathbf{B}_i^* \text{ for all } i \in S_\ell$$
$$\mathsf{sk}_{\ell,S_\ell} := (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i \in S_\ell}).$$

Game 0': This game is the same as Game 0 except for the way of making dual orthonormal bases. In Game 0, the dual orthonormal bases for the i-th element are made as $(\mathbf{B}_i, \mathbf{B}_i^*)$ where $\mathbf{B}_i := F_K(i)$, but in Game 0', they are made as $\mathbf{B}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$. More precisely, the cipertext oracle and secret key oracle have the same list $\mathcal{L}$ for bases. When the oracle needs a basis for the i-th element, it searches for $(i, \mathbf{B}_i)$ from $\mathcal{L}$. If the oracle find it, the oracle uses the bases, and if not, it generates $\mathbf{B}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$ and records them as $(i, \mathbf{B}_i)$ into $\mathcal{L}$.

Game 1-μ-1 ($\mu \in [q_{\mathsf{sk}}]$)**:** We define Game 1-0-3 as equivalent to Game 0'. This game is the same as Game 1-$(\mu-1)$-3 except that in the μ-th secret key query, $\mathbf{k}_{\mu,i}$ is set as

$$w \leftarrow \mathbb{Z}_p, \quad \mathbf{k}_{\mu,i} := (y_{\mu,i}^0, 0, r_{\mu,i}, \boxed{wr_{\mu,i}})\mathbf{B}_i^* \text{ for all } i \in S_\mu.$$

Game 1-μ-2 ($\mu \in [q_{\mathsf{sk}}]$)**:** This game is the same as Game 1-μ-1 except that in the μ-th secret key query, $\mathbf{k}_{\mu,i}$ is set as

$$w \leftarrow \mathbb{Z}_p, \quad \mathbf{k}_{\mu,i} := (y_{\mu,i}^0, \boxed{y_{\mu,i}^1}, r_{\mu,i}, wr_{\mu,i})\mathbf{B}_i^* \text{ for all } i \in S_\mu.$$

Game 1-μ-3 ($\mu \in [q_{\mathsf{sk}}]$)**:** This game is the same as Game 1-μ-2 except that in the μ-th secret key query, $\mathbf{k}_{\mu,i}$ is set as

$$\mathbf{k}_{\mu,i} := (y_{\mu,i}^0, y_{\mu,i}^1, r_{\mu,i}, \boxed{0})\mathbf{B}_i^* \text{ for all } i \in S_\mu.$$

Game 2: This game is the same as Game 1-q_{sk}-3 except that in all secret key queries, $\mathbf{k}_{\ell,i}$ for all $\ell \in [q_{sk}]$ is set as

$$\mathbf{k}_{\ell,i} := (y_{\ell,i}^0, y_{\ell,i}^1, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}})\mathbf{B}_i^* \text{ for all } i \in S_\ell,$$

where $\tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

Game 3-ν-1 ($\nu \in [q_{\mathsf{ct}}]$)**:** Game 2 is equivalent to Game 3-0-3. This game is the same as Game 3-$(\nu-1)$-3 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \leftarrow \mathbb{Z}_p, \quad \mathbf{c}_{\nu,i} := (x_{\nu,i}^0, 0, z_\nu, \boxed{\tilde{z}_\nu})\mathbf{B}_i \text{ for all } i \in [m_\nu].$$

Game 3-ν-2 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \leftarrow \mathbb{Z}_p, \quad \mathbf{c}_{\nu,i} := (\boxed{0, x^1_{\nu,i}}, z_\nu, \tilde{z}_\nu)\mathbf{B}_i \ \text{ for all } i \in [m_\nu].$$

Game 3-ν-3 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-2 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\mathbf{c}_{\nu,i} := (0, x^1_{\nu,i}, z_\nu, \boxed{0})\mathbf{B}_i \ \text{ for all } i \in [m_\nu].$$

Game 4: This game is the same as Game 3-q_{ct}-5 except that in all ciphertext and secret key queries, $\mathbf{c}_{j,i}$ and $\mathbf{k}_{\ell,i}$ are set as

$$\mathbf{c}_{j,i} := (\boxed{x^1_{j,i}, 0}, z_j, 0)\mathbf{B}_i \ \text{ for all } i \in [m_j],$$
$$\mathbf{k}_{\ell,i} := (\boxed{y^1_{\ell,i}, y^0_{\ell,i}}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i, \ \text{ for all } i \in S_\ell.$$

Game 5: This game is the same as the real security game when $\beta = 1$ in Definition 3.7. That is, the j-th ciphertext query with a pair of vectors $(\mathbf{x}^0_j, \mathbf{x}^1_j) \in (\mathbb{Z}^{m_j})^2$ is replied as

$$\mathbf{c}_{j,i} := (x^1_{j,i}, 0, z_j, 0)\boxed{\mathbf{B}_i} \ \text{ for all } i \in [m_j]$$
$$\mathsf{ct}_{j,m_j} := ([\mathbf{c}_{j,1}]_1, \ldots, [\mathbf{c}_{j,m_j}]_1).$$

The ℓ-th secret key query with an index set S_ℓ and a pair of vectors $(\mathbf{y}^0_\ell, \mathbf{y}^1_\ell) \in (\mathbb{Z}^{S_\ell})^2$ is replied as

$$\mathbf{k}_{\ell,i} := (y^1_{\ell,i}, \boxed{0}, r_{\ell,i}, \boxed{0})\boxed{\mathbf{B}^*_i} \ \text{ for all } i \in S_\ell$$
$$\mathsf{sk}_{\ell,S_\ell} := (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i \in S_\ell}).$$

Note that $\mathbf{B}_i$ is generated as $\mathbf{B}_i := F_K(i)$ in Game 5.

Thanks to Lemma 5.1 to Lemma 5.16, we can conclude the proof of Theorem 5.1. □

In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game ι by E_ι.

Lemma 5.1. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for PRFs s.t.*

$$|\mathsf{Pr}[\mathsf{E}_0] - \mathsf{Pr}[\mathsf{E}_{0'}]| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. First, we consider Game 0_{M}, which is the same as Game 0 except that $\mathbf{B}_i$ is generated as $\mathbf{B}_i \leftarrow \mathsf{M}_4(\mathbb{Z}_p)$ for each i. The following inequality directly follows from the property of PRF s.t. $|\mathsf{Pr}[\mathsf{E}_0] - \mathsf{Pr}[\mathsf{E}_{0_{\mathsf{M}}}]| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda)$. Next, we have $|\mathsf{Pr}[\mathsf{E}_{0_{\mathsf{M}}}] - \mathsf{Pr}[\mathsf{E}_{0'}]| \leq 2^{-\Omega(\lambda)}$ from Lemma 3.1. Then Lemma 5.1 holds. □

Lemma 5.2. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}(\mu-1)\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}_i, \mathbf{B}_i^*$ as follows,

$$\mathbf{W}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p), \quad \mathbf{B}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & -a \end{pmatrix} \mathbf{W}_i, \quad \mathbf{B}_i^* := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & a & 1 \\ & & 1 & 0 \end{pmatrix} \mathbf{W}_i^* \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all ciphertext queries and all secret key queries except the μ-th one as follows.

$$[\mathbf{c}_{j,i}]_1 := [(x_{j,i}^0, 0, z_{j,i}, 0)\mathbf{B}_i]_1 \text{ for all } i \in [m_j],$$
$$[\mathbf{k}_{\ell,i}]_2 := \begin{cases} [(y_{\ell,i}^0, y_{\ell,i}^1, r_{\ell,i}, 0)\mathbf{B}_i^*]_2 \text{ for all } i \in S_\ell & (\ell < \mu) \\ [(y_{\ell,i}^0, 0, r_{\ell,i}, 0)\mathbf{B}_i^*]_2 \text{ for all } i \in S_\ell & (\ell > \mu). \end{cases}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}_{i,4}]_1$ because it does not know $[a]_1$, but the above instances are computable without $[\mathbf{b}_{i,4}]_1$. For the μ-th secret key query, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in S_\mu$ as

$$r_i' \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\mu} r_i' = 0,$$
$$[\mathbf{k}_{\mu,i}]_2 := [(y_{\mu,i}^0, 0, 0, 0)\mathbf{B}_i^* + r_i'(0, 0, t_\beta, e)\mathbf{W}_i^*]_2 = [(y_{\mu,i}^0, 0, er_i', \beta f r_i')\mathbf{B}_i^*]_2.$$

Observe that we can implicitly set $r_{\mu,i} := er_i'$ and $w := f/e$ unless $e = 0$, then $\mathcal{A}$'s view is the same as in Game 1-$(\mu-1)$-3 (resp. Game 1-μ-1) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 5.3. *For any PPT adversary $\mathcal{A}$, we have*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}2}]| \leq 2^{-\Omega(\lambda)}.$$

Proof. We define $(\mathbf{D}_i, \mathbf{D}_i^*)$ as

$$\mathbf{D}_i := \begin{pmatrix} 1 & & & 0 \\ & 1 & & \frac{y_{\mu,i}^1}{wr_{\mu,i}} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \mathbf{B}_i, \quad \mathbf{D}_i^* := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 0 & -\frac{y_{\mu,i}^1}{wr_{\mu,i}} & 0 & 1 \end{pmatrix} \mathbf{B}_i^* \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}_i^*)$ are random dual orthonormal bases. Then, for all $j \in [q_{\mathsf{ct}}]$ and $\ell \in [q_{\mathsf{sk}}]$, we

have

$$\mathbf{c}_{j,i} = (x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i = (x^0_{j,i}, 0, z_{j,i}, 0)\begin{pmatrix} 1 & & & 0 \\ & 1 & & -\frac{y^1_{\mu,i}}{wr_{\mu,i}} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix}\mathbf{D}_i = (x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{D}_i,$$

$$\begin{aligned}\mathbf{k}_{\ell,i} &= (y^0_{\ell,i}, \beta_\ell y^1_{\ell,i}, r_{\ell,i}, \hat{\beta}_\ell w r_{\mu,i})\mathbf{B}^*_i = (y^0_{\ell,i}, \beta_\ell y^1_{\ell,i}, r_{\ell,i}, \hat{\beta}_\ell w r_{\mu,i})\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 0 & \frac{y^1_{\mu,i}}{wr_{\mu,i}} & 0 & 1 \end{pmatrix}\mathbf{D}^*_i \\ &= (y^0_{\ell,i}, (\beta_\ell + \hat{\beta}_\ell)y^1_{\ell,i}, r_{\ell,i}, \hat{\beta}_\ell w r_{\mu,i})\mathbf{D}^*_i,\end{aligned}$$

where $\beta_\ell = 0$ if $\ell \geq \mu$ and $\beta_\ell = 1$ if $\ell < \mu$, and $\hat{\beta}_\ell = 0$ if $\ell \neq \mu$ and $\hat{\beta}_\ell = 1$ if $\ell = \mu$. Then if $w \neq 0$ and $r_{\mu,i} \neq 0$, $\mathcal{A}$'s view is identically distributed in Game 1-μ-2 and Game 1-μ-3. □

Lemma 5.4. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}2}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

This lemma can be proven almost the same as Lemma 5.2, so we omit the proof.

Lemma 5.5. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}q_{\mathsf{sk}}\text{-}3}] - \mathsf{Pr}[\mathsf{E}_2]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}_i, \mathbf{B}^*_i$ as follows,

$$\mathbf{W}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p),\ \ \mathbf{B}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & -a \end{pmatrix}\mathbf{W}_i,\ \ \mathbf{B}^*_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & a & 1 \\ & & 1 & 0 \end{pmatrix}\mathbf{W}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all ciphertext queries and all secret key queries as follows.

$$\begin{aligned}&[\mathbf{c}_{j,i}]_1 := [(x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i]_1 \ \text{ for all } i \in [m_j], \\ &r'_{\ell,i}, r''_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ s.t. } \sum_{i \in S_\ell} r'_{\ell,i} = \sum_{i \in S_\ell} r''_{\ell,i} = 0, \\ &[\mathbf{k}_{\ell,i}]_2 := [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i}, 0)\mathbf{B}^*_i + r''_{\ell,i}(0, 0, t_\beta, e)\mathbf{W}^*_i]_2 \\ &= [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i} + er''_{\ell,i}, \beta f r''_{\ell,i})\mathbf{B}^*_i]_2 \ \text{ for all } i \in S_\ell.\end{aligned}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}_{i,4}]_1$ because it does not know $[a]_1$, but the above instances are computable without $[\mathbf{b}_{i,4}]_1$. Observe that we can implicitly set $r_{\ell,i} := r'_{\ell,i} + er''_{\ell,i}$ and $\tilde{r}_{\ell,i} := fr''_{\ell,i}$ unless $f = 0$, then $\mathcal{A}$'s view is the same as in Game 1-q_{sk}-3 (resp. Game 2) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 5.6. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}(\nu-1)\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 1$, i.e., $(\mathbb{G}, [a]_1, [e]_1, [t_\beta]_1)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}_i, \mathbf{B}_i^*$ as follows,

$$\mathbf{W}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p), \quad \mathbf{B}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & a & 1 \\ & & 1 & 0 \end{pmatrix} \mathbf{W}_i, \quad \mathbf{B}_i^* := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & -a \end{pmatrix} \mathbf{W}_i^* \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all ciphertext queries except the ν-th one and all secret key queries as follows,

$$\begin{aligned}
[\mathbf{c}_{j,i}]_1 &:= \begin{cases} [(0, x^1_{j,i}, z_{j,i}, 0)\mathbf{B}_i]_1 & \text{for all } i \in [m_j] \quad (j < \nu) \\ [(x^0_{j,i}, 0, z_{j,i}, 0)\mathbf{B}_i]_1 & \text{for all } i \in [m_j] \quad (j > \nu), \end{cases} \\
& r'_{\ell,i}, r''_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} r'_{\ell,i} = \sum_{i \in S_\ell} r''_{\ell,i} = 0, \\
[\mathbf{k}_{\ell,i}]_2 &:= [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i}, 0)\mathbf{B}_i^* + (0, 0, r''_{\ell,i}, 0)\mathbf{W}_i^*]_2 \\
&= [(y^0_{\ell,i}, y^1_{\ell,i}, r'_{\ell,i} + a r''_{\ell,i}, r''_{\ell,i})\mathbf{B}_i^*]_2 \text{ for all } i \in S_\ell.
\end{aligned}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}^*_{i,4}]_2$ because it does not know $[a]_2$, but the above instances are computable without $[\mathbf{b}^*_{i,4}]_2$. Observe that we can implicitly set $r_{\ell,i} := r'_{\ell,i} + a r''_{\ell,i}$ and $\tilde{r}_{\ell,i} := r''_{\ell,i}$, so $\mathcal{B}$ correctly simulates the answer for queries. For the ν-th ciphertext query, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in [m_\nu]$ as

$$[\mathbf{c}_{\nu,i}]_1 := [(x^0_{\nu,i}, 0, 0, 0)\mathbf{B}_i + (0, 0, t_\beta, e)\mathbf{W}_i]_1 = [(x^0_{\nu,i}, 0, e, \beta f)\mathbf{B}_i]_1.$$

Observe that we can implicitly set $z_\nu := e$ and $\tilde{z}_\nu := f$, then $\mathcal{A}$'s view is the same as in Game 3-$(\nu-1)$-3 (resp. Game 3-ν-1) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 5.7. *Let m_{max} be the maximum length of vectors with which $\mathcal{A}$ makes a query to the encryption oracle. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}]| \leq 2m_{\mathsf{max}}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. To prove Lemma 5.7, we consider the following intermediate games between Game 3-ν-1 and 3-ν-2. In each intermediate game, the challenger chooses a random element $m'_\nu \leftarrow [m_{\mathsf{max}}]$ as a guess of m_ν at the beginning of the games.

Game 3-ν-1-1 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1 except that the challenger aborts the game immediately if the vector length of the ν-th ciphertext query is not m'_ν i.e., $m'_\nu \neq m_\nu$. We define that $\mathcal{A}$'s output is $\perp$ when the game is aborted.

Game 3-ν-1-2 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-1 except the following. In the ℓ-th secret key query for all ℓ s.t. whose index set S_ℓ contains both elements that are greater than m'_ν and not greater than m'_ν, i.e., $(\max S_\ell > m'_\nu) \wedge (\min S_\ell \leq m'_\nu)$, $\mathbf{k}_{\ell,i}$ is set as

$$\mathbf{k}_{\ell,i} := \begin{cases} (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i & (i \in S_\ell, i \leq m'_\nu) \\ (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{a\tilde{r}_{\ell,i}})\mathbf{B}^*_i & (i \in S_\ell, i > m'_\nu) \end{cases}$$

where $a \leftarrow \mathbb{Z}_p, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

Game 3-ν-1-3 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-2 except that in the ℓ-th secret key query for all ℓ s.t. $(\max S_\ell > m'_\nu) \wedge (\min S_\ell \leq m'_\nu)$, $\mathbf{k}_{\ell,i}$ is set as

$$\bar{r}_{\ell,i} \leftarrow \mathbb{Z}_p, \quad \mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}})\mathbf{B}^*_i \text{ for all } i \in S_\ell.$$

Game 3-ν-1-4 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-3 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \leftarrow \mathbb{Z}_p, \quad \mathbf{c}_{\nu,i} := (\boxed{0, x^1_{\nu,i}}, z_\nu, \tilde{z}_\nu)\mathbf{B}_i \text{ for all } i \in [m'_\nu].$$

Game 3-ν-1-5 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1-4 except that in all secret key queries, $\mathbf{k}_{\ell,i}$ are set as

$$\mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}})\mathbf{B}^*_i \text{ for all } i \in S_\ell,$$

where $\tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

Next we consider the probability that $\mathcal{A}$ outputs 1 in each game. Thanks to Lemma 5.8 to Lemma 5.13, we have

$$\begin{aligned} |\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}]| &= m_{\mathsf{max}} |\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}]| \\ &\leq 2m_{\mathsf{max}} \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)} \end{aligned}$$

This concludes the proof of Lemma 5.7. □

Lemma 5.8. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] = \frac{1}{m_{\mathsf{max}}} \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}]$$

Proof. First, we consider the game (denoted by Game X) that is the same as Game 3-ν-1 except that $\mathcal{A}$'s output is defined as $\perp$ when $m'_\nu \neq m_\nu$. Note that the challenger does not abort the game in Game X in contrast to Game 3-ν-1-1. It is obvious that the probabilities that $\mathcal{A}$ outputs 1 are equal in Game X and Game 3-ν-1-1 respectively. Then, we have

$$\begin{aligned} \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] = \mathsf{Pr}[\mathsf{E}_X] &= \sum_{i \in [m_{\mathsf{max}}]} \mathsf{Pr}[m'_\nu = i] \mathsf{Pr}[m_\nu = i \wedge \mathsf{E}_{3\text{-}\nu\text{-}1} | m'_\nu = i] \\ &= \frac{1}{m_{\mathsf{max}}} \sum_{i \in [m_{\mathsf{max}}]} \mathsf{Pr}[m_\nu = i \wedge \mathsf{E}_{3\text{-}\nu\text{-}1}] \\ &= \frac{1}{m_{\mathsf{max}}} \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1}]. \end{aligned}$$

The second line follows from the fact that m'_ν is chosen independently from $\mathcal{A}$'s view in Game X and its value does not affect $\mathcal{A}$'s behavior. □

Lemma 5.9. *For any PPT adversary $\mathcal{A}$, we have*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}2}]| \leq 2^{-\Omega(\lambda)}.$$

Proof. For $i > m'_\nu$, we define $(\mathbf{D}_i, \mathbf{D}_i^*)$ as

$$\mathbf{D}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & a \end{pmatrix} \mathbf{B}_i, \quad \mathbf{D}_i^* := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/a \end{pmatrix} \mathbf{B}_i^* \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Ciphertexts except the ν-th one and secret keys that have indices greater than m'_ν are changed as

$$\begin{aligned}
\mathbf{c}_{j,i} &= (\beta_j x^0_{j,i}, (1-\beta_j) x^1_{j,i}, z_j, 0)\mathbf{B}_i = (\beta_j x^0_{j,i}, (1-\beta_j) x^1_{j,i}, z_j, 0) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/a \end{pmatrix} \mathbf{D}_i \\
&= (\beta_j x^0_{j,i}, (1-\beta_j) x^1_{j,i}, z_j, 0)\mathbf{D}_i \ \text{ for all } i > m'_\nu, \\
\mathbf{k}_{\ell,i} &= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}_i^* = (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & a \end{pmatrix} \mathbf{D}_i^* \\
&= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, a\tilde{r}_{\ell,i})\mathbf{D}_i^* \ \text{ for all } i > m'_\nu,
\end{aligned}$$

where $\beta_j = 0$ if $j < \nu$ and $\beta_j = 1$ if $j \geq \nu$. Note that secret keys whose all indices are greater than m'_ν are not affected by the basis change because $\{\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$ and $\{a\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$ are identically distributed. Finally, when $m'_\nu = m_\nu$, this basis change does not affect ct_{ν,m_ν} because it is applied only for the bases with indices $i > m_\nu$. Hence, in Game 3-ν-1-1 and Game 3-ν-1-2, $\mathcal{A}$'s view is identically distributed unless $a = 0$. □

Lemma 5.10. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}2}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. In the beginning of the simulation, $\mathcal{B}$ chooses a $m'_\nu \leftarrow [m_{\mathsf{max}}]$ as a guess of m_ν. If the guess is incorrect, $\mathcal{B}$ aborts and outputs 0. Otherwise, $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is. $\mathcal{B}$ obtains an SXDH instance with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$ and gives $\mathsf{pp} := \mathbb{G}$ to $\mathcal{A}$. $\mathcal{B}$ defines dual orthonormal bases as $\mathbf{B}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$ for each index i. Then, all ciphertexts and the ℓ-th secret key s.t. $(\max S_\ell \leq$

$m'_\nu) \vee (\min S_\ell > m'_\nu)$ can be generated by using $\mathbf{B}_i$ and $\mathbf{B}^*_i$. For the ℓ-th secret key s.t. $(\max S_\ell > m'_\nu) \wedge (\min S_\ell \leq m'_\nu)$, $\mathcal{B}$ computes secret keys as follows.

$$u_{\ell,i}, u'_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} u_{\ell,i} = \sum_{i \in S_\ell} u'_{\ell,i} = 0,$$

$$[\mathbf{k}_{\ell,i}]_2 := \begin{cases} [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, eu_{\ell,i} + u'_{\ell,i})\mathbf{B}^*]_2 & (i \in S_\ell, i \leq m'_\nu) \\ [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, t_\beta u_{\ell,i} + au'_{\ell,i})\mathbf{B}^*]_2 & \\ = [(y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, a(eu_{\ell,i} + u'_{\ell,i}) + \beta f u_{\ell,i})\mathbf{B}^*]_2 & (i \in S_\ell, i > m'_\nu) \end{cases}$$

Then, we can define $\tilde{r}_{\ell,i} := eu_{\ell,i} + u'_{\ell,i}$. In the case of $\beta = 0$, $[\mathbf{k}_{\ell,i}]_2$ is distributed identically to Game 3-ν-1-2. Next, we consider the case $\beta = 1$. First, $\{\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ and $\{u_{\ell,i}\}_{i \in S_\ell}$ are independently distributed because the information of $\{u_{\ell,i}\}_{i \in S_\ell}$ in $\{\tilde{r}_{\ell,i}\}_{i \in S_\ell}$ is completely hidden by $\{u'_{\ell,i}\}_{i \in S_\ell}$. Therefore, we can set $\bar{r}_{\ell,i} := \begin{cases} \tilde{r}_{\ell,i} & (i \in S_\ell, i \leq m'_\nu) \\ a\tilde{r}_{\ell,i} + fu_{\ell,i} & (i \in S_\ell, i > m'_\nu) \end{cases}$, unless $f = 0$. Hence, $[\mathbf{k}_{\ell,i}]_2$ is distributed identically to Game 3-ν-1-3 if $\beta = 1$. □

Lemma 5.11. *For any PPT adversary $\mathcal{A}$, we have*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}]| \leq 2^{-\Omega(\lambda)}.$$

Proof. Here, we denote the event such that $m'_\nu = m_\nu$ in Game ι by X_ι. By the game definition, we have

$$\begin{aligned} &|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}]| \\ =&|\mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}]\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}]\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}]| \\ =&|\mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}](\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}])|. \end{aligned}$$

In the third line, we use the fact that $\mathcal{A}$'s view is identical before the ν-th ciphertext query, and we have $\mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] = \mathsf{Pr}[\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}]$. Therefore, it is sufficient to prove that $|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}|\mathsf{X}_{3\text{-}\nu\text{-}1\text{-}4}])| \leq 2^{-\Omega(\lambda)}$. For the purpose, we analyze $\mathcal{A}$'s view under the condition such that $m'_\nu = m_\nu$.

We define $(\mathbf{D}_i, \mathbf{D}^*_i)$ for all $i \in [m_\nu]$ as

$$\mathbf{D}_i := \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ \frac{x^0_{\nu,i}}{\tilde{z}_\nu} & -\frac{x^1_{\nu,i}}{\tilde{z}_\nu} & 0 & 1 \end{pmatrix} \mathbf{B}_i, \quad \mathbf{D}^*_i := \begin{pmatrix} 1 & & -\frac{x^0_{\nu,i}}{\tilde{z}_\nu} \\ & 1 & \frac{x^1_{\nu,i}}{\tilde{z}_\nu} \\ & & 1 \end{pmatrix} \mathbf{B}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}_i^*)$ are random dual orthonormal bases. Then, for all $j \in [q_{\mathsf{ct}}]$, we have

$$\begin{aligned}
\mathbf{c}_{j,i} &= (\beta_j x^0_{j,i}, (1-\beta_j) x^1_{j,i}, z_j, \hat{\beta}_j \tilde{z}_\nu)\mathbf{B}_i \\
&= (\beta_j x^0_{j,i}, (1-\beta_j) x^1_{j,i}, z_j, \hat{\beta}_j \tilde{z}_\nu) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ -\frac{x^0_{\nu,i}}{\tilde{z}_\nu} & \frac{x^1_{\nu,i}}{\tilde{z}_\nu} & 0 & 1 \end{pmatrix} \mathbf{D}_i \\
&= ((\beta_j - \hat{\beta}_j) x^0_{j,i}, (1-\beta_j+\hat{\beta}_j) x^1_{j,i}, z_j, \hat{\beta}_j \tilde{z}_\nu)\mathbf{D}_i,
\end{aligned}$$

where $\beta_j = 0$ if $j < \nu$ and $\beta_j = 1$ if $j \geq \nu$, and $\hat{\beta}_j = 0$ if $j \neq \nu$ and $\hat{\beta}_j = 1$ if $j = \nu$. On the other hand, for all ℓ s.t. $\max S_\ell \leq m_\nu$, we have

$$\begin{aligned}
\mathbf{k}_{\ell,i} &= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i = (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}) \begin{pmatrix} 1 & & & \frac{x^0_{\nu,i}}{\tilde{z}_\nu} \\ & 1 & & -\frac{x^1_{\nu,i}}{\tilde{z}_\nu} \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \mathbf{D}^*_i \\
&= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i} y^0_{\ell,i} - x^1_{\nu,i} y^1_{\ell,i}))\mathbf{D}^*_i.
\end{aligned}$$

Here, we have the condition Eq. (3.6) s.t. $\sum_{i \in S_\ell}(x^0_{\nu,i} y^0_{\ell,i} - x^1_{\nu,i} y^1_{\ell,i}) = 0$, because $S_\ell \subseteq [m_\nu]$. Hence, we can set $\tilde{r}'_{\ell,i} := \tilde{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i} y^0_{\ell,i} - x^1_{\nu,i} y^1_{\ell,i})$. Observe that $\tilde{r}'_{\ell,i}$ is randomly distributed s.t. $\sum_{i \in S_\ell} \tilde{r}'_{\ell,i} = 0$. In the same way, for all ℓ s.t. $(\max S_\ell > m_\nu) \wedge (\min S_\ell \leq m_\nu)$, we have

$$\mathbf{k}_{\ell,i} = \begin{cases} (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i} y^0_{\ell,i} - x^1_{\nu,i} y^1_{\ell,i}))\mathbf{D}^*_i & (i \leq m_\nu) \\ (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i})\mathbf{B}^*_i & (i > m_\nu) \end{cases}$$

In this case, there is no condition on $(x^0_{\nu,i} y^0_{\ell,i} - x^1_{\nu,i} y^1_{\ell,i})$. However, because $\bar{r}_{\ell,i}$ are chosen randomly from $\mathbb{Z}_p$, then $\bar{r}'_{\ell,i} := \bar{r}_{\ell,i} + \frac{1}{\tilde{z}_\nu}(x^0_{\nu,i} y^0_{\ell,i} - x^1_{\nu,i} y^1_{\ell,i})$ are also random elements in $\mathbb{Z}_p$. Note that for all ℓ s.t. $\min S_\ell > m_\nu$, this basis change does not affect $\mathsf{sk}_{\ell,S_\ell}$ because we only change the bases for $i \leq m_\nu$. Then, in Game 3-ν-1-3 and Game 3-ν-1-4, $\mathcal{A}$'s view is identically distributed unless $\tilde{z}_\nu = 0$ under the condition such that $m'_\nu = m_\nu$. □

Lemma 5.12. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Lemma 5.12 can be proven by just the reverse of Game 3-$(\nu-1)$-1-1 to Game 3-ν-1-3, so we omit the proof.

Lemma 5.13. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}] = \frac{1}{m_{\mathsf{max}}}\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}]$$

The difference between Game 3-ν-1-5 and 3-ν-2 is just the existence of the abort condition introduced in Game 3-ν-1-1. Then, we can prove Lemma 5.13 similarly to Lemma 5.8.

Lemma 5.14. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}2}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

This lemma can be proven by just the reverse of Game 3-$(\nu-1)$-3 to Game 3-ν-1, so we omit the proof.

Lemma 5.15. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_{3\text{-}q_{\mathsf{ct}}\text{-}3}] = \mathsf{Pr}[\mathsf{E}_4].$$

Proof. We define $(\mathbf{D}_i, \mathbf{D}^*_i)$ as

$$\mathbf{D}_i := \begin{pmatrix} & 1 & & \\ 1 & & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \mathbf{B}_i, \quad \mathbf{D}^*_i := \begin{pmatrix} & 1 & & \\ 1 & & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \mathbf{B}^*_i \in \mathsf{GL}_4(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}^*_i)$ are random dual orthonormal bases. Then, for all $j \in [q_{\mathsf{ct}}]$ and $\ell \in [q_{\mathsf{sk}}]$, we have

$$\begin{aligned} \mathbf{c}_{j,i} &= (0, x^1_{j,i}, z_j, 0)\mathbf{B}_i = (x^1_{j,i}, 0, z_j, 0)\mathbf{D}_i \ \text{ for all } i \in [m_j], \\ \mathbf{k}_{\ell,i} &= (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i = (y^1_{\ell,i}, y^0_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{D}^*_i \ \text{ for all } i \in S_\ell. \end{aligned}$$

Then, in Game 3-q_{ct}-3 and Game 4, $\mathcal{A}$'s view is identically distributed. □

Lemma 5.16. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_1$ for the SXDH and $\mathcal{B}_2$ for PRF s.t.*

$$|\mathsf{Pr}[\mathsf{E}_4] - \mathsf{Pr}[\mathsf{E}_5]| \leq (2q_{\mathsf{sk}} + 1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)}.$$

This lemma can be proven by just the reverse of Games 0 to 2, so we omit the proof.

5.2.3 Selectively Function-Hiding Scheme for (E:sep, K:sep, D:ct-dom)

In this section, we explain our selectively function-hiding Priv-UIPFE scheme for the (E:sep, K:sep, D:ct-dom) setting, of which the function class is defined as follows.

Definition 5.1 (Bounded-Norm Unbounded Inner Product over $\mathbb{Z}$ (E:sep, K:sep, D:ct-dom))**.** This function family $\mathcal{F}^{\mathsf{UIP2}}_{X,Y}$, where $X, Y \in \mathbb{N}$, consists of functions $f_{S,\mathbf{y}} : \mathcal{X} \to \mathbb{Z} \cup \{\perp\}$, where $S \subset \mathbb{N}$, $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$ s.t. $||\mathbf{y}||_\infty \leq Y$, and $\mathcal{X} = \{(\mathbf{x}, U) \mid (\mathbf{x}, U) \in \bigcup_{i \in \mathbb{N}} \mathbb{Z}^i \times \mathcal{U}_i, ||\mathbf{x}||_\infty \leq X\}$, where $\mathcal{U}_i = \{U \mid U \subset \mathbb{N}, |U| = i\}$. We define the function for all $(\mathbf{x}, U) \in \mathcal{X}$ as

$$f_{S,\mathbf{y}}(\mathbf{x}, U) := \begin{cases} \sum_{i \in S} x_i y_i & (S \subseteq U) \\ \perp & (S \nsubseteq U) \end{cases}.$$

The syntax of UIPFE for (E:sep, K:sep, D:ct-dom) is the same as that of UIPFE for (E:con, K:sep, D:ct-dom) except that vectors to be encrypted can be a separate one. More precisely, a vector to be encrypted has a form such that $\mathbf{x} := (x_i)_{i\in U}$ for a index set U rather than $\mathbf{x} := (x_i)_{i\in[m]}$. The construction of our selectively function-hiding scheme for (E:sep, K:sep, D:ct-dom) is the same as our E:con scheme (Section 5.2.1) except that the form of vectors to be encrypted is $\mathbf{x} := (x_i)_{i\in U}$ rather than $\mathbf{x} := (x_i)_{i\in[m]}$, where $U \subseteq [u]$ for any polynomial $u := u(\lambda)$. The correctness holds in the same manner as our E:con scheme. The security statement is somewhat different from that of our E:con scheme as follows.

Theorem 5.2. *Assume that the SXDH assumption holds and $\mathcal{F}$ is a PRF family, then our Priv-UIPFE is selectively function-hiding in the (E:sep, K:sep, D:ct-dom) setting. More formally, for any PPT adversary $\mathcal{A}$ and security parameter λ, there exists a PPT adversary $\mathcal{B}_1$ for the SXDH and $\mathcal{B}_2$ for the PRF family, we have*

$$\mathsf{Adv}^{\mathsf{Priv\text{-}UIPFE}}_{\mathcal{A}}(\lambda) \leq (4q_{\mathsf{sk}} + 4q_{\mathsf{ct}} + 2)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}_1}(\lambda) + 2\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. The proof of Theorem 5.2 is almost the same as that of Theorem 5.1 except the Game 3 sequence. That is, instead of guessing the index set of the ν-th ciphertext between Game 3-ν-1 and 3-ν-2, which incurs exponential security loss in the adaptive E:sep setting, the reduction obtains the index set before it generates the secret keys in the selective setting. Concretely, the game transition is changed to the following way.

Game 3-ν-1 ($\nu \in [q_{\mathsf{ct}}]$)**:** Game 2 is equivalent to Game 3-0-5. This game is the same as Game 3-($\nu-1$)-5 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$\tilde{z}_\nu \leftarrow \mathbb{Z}_p,\ \ \mathbf{c}_{\nu,i} := (x^0_{\nu,i}, 0, z_\nu, \boxed{\tilde{z}_\nu})\mathbf{B}_i \ \text{ for all } i \in U_\nu.$$

Game 3-ν-2 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-1 except the following. In the ℓ-th secret key query for all ℓ s.t. whose index set S_ℓ contains both elements that are contained in U_ν and not contained in U_ν, i.e., $(S_\ell \cap U_\nu \neq \phi) \wedge (S_\ell \backslash U_\nu \neq \phi)$, $\mathbf{k}_{\ell,i}$ is set as

$$\mathbf{k}_{\ell,i} := \begin{cases} (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i})\mathbf{B}^*_i & (i \in S_\ell, i \in U_\nu) \\ (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{a\tilde{r}_{\ell,i}})\mathbf{B}^*_i & (i \in S_\ell, i \notin U_\nu) \end{cases}$$

where $a \leftarrow \mathbb{Z}_p, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i\in S_\ell} \tilde{r}_{\ell,i} = 0$.

Game 3-ν-3 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-2 except that in the ℓ-th secret key query for all ℓ s.t. $(S_\ell \cap U_\nu \neq \phi) \wedge (S_\ell \backslash U_\nu \neq \phi)$, $\mathbf{k}_{\ell,i}$ is set as

$$\bar{r}_{\ell,i} \leftarrow \mathbb{Z}_p,\ \ \mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}})\mathbf{B}^*_i \ \text{ for all } i \in S_\ell.$$

Game 3-ν-4 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-3 except that in the ν-th ciphertext query, $\mathbf{c}_{\nu,i}$ is set as

$$v \leftarrow \mathbb{Z}_p,\ \ \mathbf{c}_{\nu,i} := (\boxed{0, x^1_{\nu,i}}, z_\nu, \tilde{z}_\nu)\mathbf{B}_i \ \text{ for all } i \in U_\nu.$$

Game 3-ν-5 ($\nu \in [q_{\mathsf{ct}}]$)**:** This game is the same as Game 3-ν-4 except that in the ν-th ciphertext query and all secret key queries, $\mathbf{c}_{\nu,i}$ and $\mathbf{k}_{\ell,i}$ are set as

$$\mathbf{c}_{\nu,i} := (0, x^1_{\nu,i}, z_\nu, \boxed{0})\mathbf{B}_i \ \text{ for all } i \in U_\nu,$$
$$\mathbf{k}_{\ell,i} := (y^0_{\ell,i}, y^1_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}})\mathbf{B}^*_i \ \text{ for all } i \in S_\ell,$$

where $\tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0$.

In the E:sep setting, we can classify secret keys into three types for ν-th ciphertext in the same way as the E:con setting. Namely,

1. The index set S of the secret key is included in U_ν, i,e., $S \subseteq U_\nu$.
2. A part of the index set S is included in U_ν, i.e., $(S_\ell \cap U_\nu \neq \phi) \wedge (S_\ell \backslash U_\nu \neq \phi)$.
3. The index set S and U_ν are disjoint, i.e., $S \cap U_\nu = \phi$.

Observe that the way of the classification is the same as the E:con case. In addition, proofs of lemmas in the game transition are almost the same because the treatment of each type of keys is not changed. Note that the reduction from the difference between Game 3-ν-1 and 3-ν-2 to the SXDH problem in the E:con case does not need guess. Then we have

$$|\mathsf{Pr}[\mathsf{E}_{3\text{-}(\nu-1)\text{-}5}] - \mathsf{Pr}[\mathsf{E}_{3\text{-}\nu\text{-}5}]| \leq 4\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

□

5.2.4 Fully Function-Hiding Scheme for (E:sep, K:sep, D:eq)

In this section, we explain our fully function-hiding Priv-UIPFE scheme for the (E:sep, K:sep, D:eq) setting, of which the function class is defined as follows.

Definition 5.2 (Bounded-Norm Unbounded Inner Product over $\mathbb{Z}$ (E:sep, K:sep, D:eq))**.** This function family $\mathcal{F}^{\mathsf{UIP3}}_{X,Y}$, where $X, Y \in \mathbb{N}$, consists of functions $f_{S,\mathbf{y}} : \mathcal{X} \to \mathbb{Z} \cup \{\perp\}$, where $S \subset \mathbb{N}$, $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$ s.t. $||\mathbf{y}||_\infty \leq Y$, and $\mathcal{X} = \{(\mathbf{x}, U) \mid (\mathbf{x}, U) \in \bigcup_{i \in \mathbb{N}} \mathbb{Z}^i \times \mathcal{U}_i, ||\mathbf{x}||_\infty \leq X\}$, where $\mathcal{U}_i = \{U \mid U \subset \mathbb{N}, |U| = i\}$. We define the function for all $(\mathbf{x}, U) \in \mathcal{X}$ as

$$f_{S,\mathbf{y}}(\mathbf{x}, U) := \begin{cases} \sum_{i \in S} x_i y_i & (S = U) \\ \perp & (S \neq U) \end{cases}.$$

Construction. In the following scheme, norm limits X_λ, Y_λ are some polynomials. Let $\mathcal{F} := \{F_K\}_{K \in \mathcal{K}_\lambda}$ be a PRF family with a key space $\mathcal{K}_\lambda$ consisting of functions $F_K : \{0,1\}^* \to \mathbb{Z}_p$. Note that a PRF family with variable length is constructible from one-way functions in the standard model [BR05].

$\mathsf{Setup}(1^\lambda)$**:** Takes a security parameter 1^λ and chooses bilinear groups $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$ a PRF key $K \leftarrow \mathcal{K}_\lambda$. Outputs

$$\mathsf{pp} := \mathbb{G}, \ \ \mathsf{msk} := K.$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, U, \mathbf{x})$**:** Takes $\mathsf{pp}, \mathsf{msk}, U$, and $\mathbf{x} := (x_i)_{i \in U} \in \mathbb{Z}^U$. Sets $b_{i,j} := F_K(U||\$||i(2|U|+5)+j)$, $\mathbf{B}_U := (b_{i,j})_{i,j \in [2|U|+5]} \in \mathsf{M}_{2|U|+5}(\mathbb{Z}_p)$, and $\mathbf{c}_U := (\mathbf{x}, 0^{|U|}, \mathbf{r}, 0, 0, 0)\mathbf{B}_U \in \mathbb{Z}_p^{2|U|+5}$ where $\mathbf{r} \leftarrow \mathbb{Z}_p^2$. Outputs

$$\mathsf{ct}_U := (U, [\mathbf{c}_U]_1).$$

If $\mathbf{B}_U$ is a singular matrix, outputs $\perp$. Note that \$ is a special symbol that is not used to encode U and $i(2|U|+5)+j$.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, S, \mathbf{y})$**:** Takes $\mathsf{pp}, \mathsf{msk}, S$, and $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$. Sets $b_{i,j} := F_K(S||\$||i(2|S|+5)+j)$ and $\mathbf{B}_S := (b_{i,j})_{i,j \in [2|S|+5]} \in \mathsf{M}_{2|S|+5}(\mathbb{Z}_p)$. Then computes $\mathbf{k}_S := (\mathbf{y}, 0^{|S|}, 0, 0, \mathbf{s}, 0)\mathbf{B}_S^* \in \mathbb{Z}_p^{2|S|+5}$ where $\mathbf{s} \leftarrow \mathbb{Z}_p^2$. Outputs

$$\mathsf{sk}_S := (S, [\mathbf{k}_S]_2).$$

If $\mathbf{B}_S$ is a singular matrix, outputs $\perp$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}_U, \mathsf{sk}_S)$**:** Takes pp, a ciphertext ct_U, and a secret key sk_S. If $U = S$, then computes

$$h := e([\mathbf{c}_U]_1, [\mathbf{k}_S]_2),$$

and searches for d s.t. $e(g_1, g_2)^d = h$ exhaustively in the range of $-|U|X_\lambda Y_\lambda$ to $|U|X_\lambda Y_\lambda$. If such d is found, outputs d. Otherwise, outputs $\perp$.

Correctness. This scheme is correct if $\mathcal{F}$ is a PRF family. We consider the case where $\mathbf{B}_U$ is invertible. Observe that if $U = S$,

$$h = e([\mathbf{c}_U]_1, [\mathbf{k}_S]_2) = e(g_1, g_2)^{\sum_{i \in U} x_i y_i}.$$

If $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, then $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq |U|X_\lambda Y_\lambda$ and Dec outputs $\sum_{i \in U} x_i y_i$. Hence, if $\mathbf{B}_U$ is invertible without a negligible probability, our scheme is correct. By the similar logic to our Priv-UIPFE scheme in Section 5.2.1, the above statement holds.

Security. We briefly explain the proof idea. The challenger first changes the way to generate a matrix $\mathbf{B}_U$ as $\mathbf{B}_U \leftarrow \mathsf{GL}_{2|U|+5}(\mathbb{Z}_p)$. A PPT adversary cannot distinguish this change similarly to Lemma 5.1. At this point, the situation is the same as one where the fully function hiding bounded IPFE schemes by Tomida *et al.* [TAO16] are executed in parallel for each index set. Hence, the security of the above scheme is reduced to that of their scheme.

5.3 Public-Key Unbounded Inner Product Functional Encryption

We present our main public-key unbounded IPFE scheme, that is, a FE scheme for $\mathcal{F}_{X,Y}^{\mathsf{UIP}}$. In the following scheme, norm limits X_λ, Y_λ are some polynomials.

5.3.1 Construction

$\mathsf{Setup}(1^\lambda)$**:** Takes a security parameter 1^λ and generates $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$ and $\mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$. Outputs

$$\mathsf{pk} := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1), \quad \mathsf{msk} := (\mathbf{b}_1^*, \ldots, \mathbf{b}_4^*),$$

where $\mathbf{b}_i$ (resp. $\mathbf{b}_j^*$) denotes the i-th row of $\mathbf{B}$ (resp. j-th row of $\mathbf{B}^*$).

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** Takes pk and $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{Z}^m$ where $m = m(\lambda)$ is any polynomial. Defines $\mathbf{c}_i := (\pi_i(1, i), x_i, z, 0, 0, 0)\mathbf{B} \in \mathbb{Z}_p^7$ for all $i \in [m]$, where $\pi_i, z \leftarrow \mathbb{Z}_p$. Outputs

$$\mathsf{ct}_m := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_m]_1).$$

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, S, \mathbf{y})$**:** Takes pk, msk, a non-empty index set $S \subseteq [s]$ where $s = s(\lambda)$ is any polynomial, and an indexed vector $\mathbf{y} := (y_i)_{i \in S} \in \mathbb{Z}^S$. Chooses $\{r_i\}_{i \in S} \leftarrow \mathbb{Z}_p$ s.t. $\sum_{i \in S} r_i = 0$ and $\rho_i \leftarrow \mathbb{Z}_p$, and defines $\mathbf{k}_i := (\rho_i(-i, 1), y_i, r_i, 0, 0, 0)\mathbf{B}^* \in \mathbb{Z}_p^7$ for all $i \in S$. Outputs

$$\mathsf{sk}_S := (S, \{[\mathbf{k}_i]_2\}_{i \in S}).$$

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_m, \mathsf{sk}_S)$**:** Takes pk, a ciphertext ct_m for m dimensional vector, and a secret key sk_S for a index set S. If $S \subseteq [m]$, then computes

$$h := \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2),$$

and searches for d s.t. $e(g_1, g_2)^d = h$ exhaustively in the range of $-|S|X_\lambda Y_\lambda$ to $|S|X_\lambda Y_\lambda$. If such d is found, outputs d. Otherwise, outputs $\perp$.

Correctness. Observe that if $S \subseteq [m]$,

$$h = \prod_{i \in S} e([\mathbf{c}_i]_1, [\mathbf{k}_i]_2) = e(g_1, g_2)^{\sum_{i \in S} \langle \mathbf{c}_i, \mathbf{k}_i \rangle} = e(g_1, g_2)^{\sum_{i \in S} (x_i y_i + z r_i)}.$$

Here we have $\sum_{i \in S} r_i = 0$, then $h = e(g_1, g_2)^{\sum_{i \in S} x_i y_i}$. If $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, then $|\sum_{i \in S} x_i y_i| \leq |S|X_\lambda Y_\lambda$ and Dec outputs $\sum_{i \in S} x_i y_i$.

5.3.2 Security

Theorem 5.3. *Assume that the SXDH assumption holds, then our Pub-UIPFE is adaptively secure. More formally, let m_{max} be the maximum length of the challenge vector that $\mathcal{A}$ outputs and s_{max} be the maximum index with which $\mathcal{A}$ queries the key generation oracle, then for any PPT adversary $\mathcal{A}$ and security parameter λ, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Pub\text{-}UIPFE}}(\lambda) \leq \{16m_{\mathsf{max}}^2 + 8m_{\mathsf{max}}(s_{\mathsf{max}} - 1) + 4\}\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof Outline. The top-level strategy of the security proof is simple. Consider a world where an encryption algorithm could magically generate unbounded random dual orthonormal bases for each index. Then we observe that only one loop of the Game 3 sequence in the Priv-UIPFE scheme suffices for the Pub-UIPFE scheme because there is one challenge ciphertext query and no challenge secret key query. To generate such a situation, we utilize an entropy-amplification technique like [OT12b] and show that PPT adversaries cannot distinguish the real world from the "magical" world under the SXDH assumption. In the following, we provide a more concrete overview of the proof. Similarly to the Game 3 sequence in the Priv-UIPFE scheme, we first change the challenge ciphertext and all secret keys into the following form,

$$\tilde{z}, \{\tilde{r}_{\ell,i}\}_{i\in S_\ell} \leftarrow \mathbb{Z}_p \ \text{ s.t. } \sum_{i\in S_\ell} \tilde{r}_{\ell,i} = 0,$$
$$\mathbf{c}_i := (\pi_i(1,i), x_i^0, z, \boxed{\tilde{z}}, 0, 0)\mathbf{B}, \ \ \mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}}, 0, 0)\mathbf{B}^*.$$

Next, we change $\mathbf{k}_{\ell,i}$ for all ℓ s.t. $(\max S_\ell > m') \wedge (\min S_\ell \le m')$, where m' is the guess of the vector length for the challenge ciphertext, as

$$\{\bar{r}_{\ell,i}\}_{i\in S_\ell} \leftarrow \mathbb{Z}_p, \ \ \mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}}, 0, 0)\mathbf{B}^*. \tag{5.1}$$

Then, we change $\mathbf{c}_i$ as

$$\mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^1}, z, \tilde{z}, 0, 0)\mathbf{B}, \tag{5.2}$$

similar to Priv-UIPFE. The remaining sequence is just the reverse. In the case of the Priv-UIPFE scheme, recall that we perform distinct basis changes for each index in the steps of Eq. (5.1) and Eq. (5.2). However, we cannot perform such basis changes in Pub-UIPFE, because all indices share the same dual orthonormal bases. To overcome this difficulty, we conduct this step by computational change on the basis of the SXDH assumption. Specifically, we introduce the Lemmas 5.17 and 5.18 and use them in the proof as a kind of basis change in Priv-UIPFE. Especially, it is relatively easy to see that Lemma 5.18 can be used for showing that PPT adversaries cannot distinguish the real world, i.e., $\beta = 0$, from the "magical" world, i.e., $\beta = 1$, where dual orthonormal bases for each index are "somewhat" random. In other words, in the case of $\beta = 1$, dual orthonormal bases for index i is generated as

$$\mathbf{D}_i := \begin{pmatrix} \mathbf{I}_2 & & & \\ & 1 & & \\ & & 1 & \\ & w_i & & 1 \\ & & & & \mathbf{I}_2 \end{pmatrix} \mathbf{B}, \ \ \mathbf{D}_i^* := \begin{pmatrix} \mathbf{I}_2 & & & \\ & 1 & & -w_i \\ & & 1 & \\ & & & 1 \\ & & & & \mathbf{I}_2 \end{pmatrix} \mathbf{B}^*. \tag{5.3}$$

Lemma 5.17 is used for the step of Eq. (5.1), which corresponds to Games 3-ν-2 and 3-ν-3 in the proof of Priv-UIPFE, and Lemma 5.18 is used for the step of Eq. (5.2), which corresponds to Game 3-ν-4 in the proof of Priv-UIPFE. In our Pub-UIPFE scheme, there are three-dimensional subspaces that are not used in the actual function: the 5-7th spaces. The fifth space is a kind of a semi-functional space that is similar to the fourth space of our Priv-UIPFE scheme. The sixth

and seventh spaces are necessary to amplify the entropy of the two dimensional prefix for the proof of the lemmas. Similar to here, adding extra spaces other than the semi-functional space and amplifying the entropy in the space are also done in [LW11, OT12b, CGKW18].

Lemma 5.17. *For any polynomial $m := m(\lambda)$ and $n := n(\lambda)$, we define the following distribution,*

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \quad \{\pi_i\}_{i\in[m]}, \tilde{z} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}_i := (\pi_i(1,i), 0, 0, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in [m],\\
&D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}_1^*]_2, \ldots, [\mathbf{b}_5^*]_2, [\mathbf{u}_1]_1, \ldots, [\mathbf{u}_m]_1),\\
&\{\rho_i'\}_{i\in[m+1,n]}, \{r_i'\}_{i\in[m+1,n]} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}_{i,\beta}^* := (\rho_i'(-i,1), 0, 0, \beta r_i', 0, 0)\mathbf{B}^* \text{ for all } i \in [m+1,n],\\
&U_\beta := \{[\mathbf{u}_{i,\beta}^*]_2\}_{i\in[m+1,n]}.
\end{aligned}$$

For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P1}}(\lambda) := |\mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_0)] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_1)]| \leq \quad 4(n-m)\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Lemma 5.18. *For any polynomial $m := m(\lambda)$ and $n := n(\lambda)$, we define the following distribution,*

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \quad \{\rho_i'\}_{i\in[m+1,n]} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}_i^* := (\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* \text{ for all } i \in [m+1,n],\\
&D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}_1^*]_2, [\mathbf{b}_2^*]_2, [\mathbf{b}_4^*]_2, [\mathbf{b}_5^*]_2, \{[\mathbf{u}_i^*]_2\}_{i\in[m+1,n]}),\\
&\{\pi_i'\}_{i\in[m]}, \{\rho_i'\}_{i\in[m]}, \{w_i\}_{i\in[m]} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}_{i,\beta} := (\pi_i'(1,i), \beta w_i, 0, 1, 0, 0)\mathbf{B} \text{ for all } i \in [m],\\
&\mathbf{u}_{i,\beta}^* := (\rho_i'(-i,1), 1, 0, -\beta w_i, 0, 0)\mathbf{B}^* \text{ for all } i \in [m],\\
&U_\beta := \{[\mathbf{u}_{i,\beta}]_1, [\mathbf{u}_{i,\beta}^*]_2\}_{i\in[m]}.
\end{aligned}$$

For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P2}}(\lambda) := |\mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_0)] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_1)]| \leq 8m\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

The proofs of Lemmas 5.17 and 5.18 are presented in Section 5.3.3.

Proof of Theorem 5.3. We prove Theorem 5.3 by a series of games. For each game transition, we evaluate the probabilities that the adversary $\mathcal{A}$ outputs 1 in both games. In each game, the challenger chooses a random element $m' \leftarrow [m_{\mathsf{max}}]$ as a guess of m^* at the beginning of the games.

Game 0: This game is the same as the real security game where the challenge ciphertext is the encryption of $\mathbf{x}^0$ as described in Definition 3.6. That is, the challenge ciphertext for a pair of vectors $(\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^{m^*})^2$ is replied as

$$\begin{aligned}
&\mathbf{c}_i := (\pi_i(1,i), x_i^0, z, 0, 0, 0)\mathbf{B} \text{ for all } i \in [m^*]\\
&\mathsf{ct}_{m^*} := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_{m^*}]_1).
\end{aligned}$$

The ℓ-th secret key query with an index set S_ℓ and vector $\mathbf{y}_\ell \in \mathbb{Z}^{S_\ell}$ is replied as

$$\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, 0, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell$$
$$\mathsf{sk}_{\ell,S_\ell} := (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i\in S_\ell}).$$

Game 1: This game is the same as Game 0 except that $\mathbf{c}_i$ in the challenge ciphertext is set as

$$\tilde{z} \leftarrow \mathbb{Z}_p, \quad \mathbf{c}_i := (\pi_i(1,i), x_i^0, z, \boxed{\tilde{z}}, 0, 0)\mathbf{B} \text{ for all } i \in [m^*].$$

Game 2: This game is the same as Game 1 except that in all secret key queries, $\mathbf{k}_{\ell,i}$ for all $\ell \in [q_{\mathsf{sk}}]$ is set as

$$\tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i\in S_\ell} \tilde{r}_{\ell,i} = 0,$$
$$\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}}, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell.$$

Game 3: This game is the same as Game 2 except that the challenger aborts the game immediately if the vector length of the ν-th ciphertext query is not m' i.e., $m' \neq m^*$. We define that $\mathcal{A}$'s output is $\perp$ when the game is aborted.

Game 4: This game is the same as Game 3 except that in the ℓ-th secret key query for all ℓ s.t. $(\max S_\ell > m') \wedge (\min S_\ell \leq m')$, $\mathbf{k}_{\ell,i}$ is set as

$$\bar{r}_{\ell,i} \leftarrow \mathbb{Z}_p,$$
$$\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}}, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell.$$

Game 5: This game is the same as Game 4 except that $\mathbf{c}_i$ and $\mathbf{k}_{\ell,i}$ in the challenge ciphertext and the ℓ-th secret key for all ℓ s.t. $\min S_\ell \leq m'$ are generated as

$$w_i, \tilde{r}_{\ell,i}, \bar{r}_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i\in S_\ell} \tilde{r}_{\ell,i} = 0,$$
$$\mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^0 + w_i\tilde{z}}, z, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in [m'],$$
$$\mathbf{k}_{\ell,i} := \begin{cases} (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i} - w_i y_{\ell,i}}, 0, 0)\mathbf{B}^* & (i \in [m'], \max S_\ell \leq m') \\ (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i} - w_i y_{\ell,i}}, 0, 0)\mathbf{B}^* & (i \in [m'], \max S_\ell > m') \end{cases}.$$

Observe that the above instances also can be written as

$$\mathbf{c}_i := (\pi_i(1,i), x_i^0, z, \tilde{z}, 0, 0)\boxed{\mathbf{D}_i} \text{ for all } i \in [m'],$$
$$\mathbf{k}_{\ell,i} := \begin{cases} (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\boxed{\mathbf{D}_i^*} & (i \in [m'], \max S_\ell \leq m') \\ (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i}, 0, 0)\boxed{\mathbf{D}_i^*} & (i \in [m'], \max S_\ell > m'), \end{cases}$$

where $\mathbf{D}_i$ and $\mathbf{D}_i^*$ are the same as Eq. (5.3).

Game 6: This game is the same as Game 5 except that $\mathbf{c}_i$ in the challenge ciphertext is set as

$$w_i, \tilde{z} \leftarrow \mathbb{Z}_p, \quad \mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^1 + w_i\tilde{z}}, z, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in [m'].$$

Game 7: This game is the same as Game 6 except that in challenge ciphertext and all secret key queries, $\mathbf{k}_{\ell,i}$ for all $\ell \in [q_{\mathsf{sk}}]$ is set as

$$\begin{aligned}
&\tilde{z}, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0, \\
&\mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^1}, z, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in [m'] \\
&\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i}}, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell.
\end{aligned}$$

Game 8: This game is the same as Game 7 except that the abort condition defined in Game 3 is removed.

Game 9: This game is the same as the real security game where the challenge ciphertext is the encryption of $\mathbf{x}^1$ as described in Definition 3.6. That is, the challenge ciphertext for a pair of vectors $(\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^{m^*})^2$ is replied as

$$\begin{aligned}
&\mathbf{c}_i := (\pi_i(1,i), x_i^1, z, \boxed{0}, 0, 0)\mathbf{B} \text{ for all } i \in [m^*] \\
&\mathsf{ct}_{m^*} := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_{m^*}]_1).
\end{aligned}$$

The ℓ-th secret key query with an index set S_ℓ and vector $\mathbf{y}_\ell \in \mathbb{Z}^{S_\ell}$ is replied as

$$\begin{aligned}
&\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{0}, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell \\
&\mathsf{sk}_{\ell,S_\ell} := (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i \in S_\ell}).
\end{aligned}$$

Thanks to Lemma 5.19 to Lemma 5.27, we can conclude the proof of Theorem 5.3. □

In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game ι by E_ι,

Lemma 5.19. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_0] - \mathsf{Pr}[\mathsf{E}_1]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 1$, i.e., $(\mathbb{G}, [a]_1, [e]_1, [t_\beta]_1)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}, \mathbf{B}^*$ as follows,

$$\mathbf{W} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \quad \mathbf{B} := \begin{pmatrix} \mathbf{I}_3 & & \\ & \begin{matrix} a & 1 \\ 1 & 0 \end{matrix} & \\ & & \mathbf{I}_2 \end{pmatrix} \mathbf{W}, \quad \mathbf{B}^* := \begin{pmatrix} \mathbf{I}_3 & & \\ & \begin{matrix} 0 & 1 \\ 1 & -a \end{matrix} & \\ & & \mathbf{I}_2 \end{pmatrix} \mathbf{W}^* \in \mathsf{GL}_7(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates all secret key queries as follows.

$$[\mathbf{k}_{\ell,i}]_2 := [(\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, 0, 0, 0)\mathbf{B}^*]_2 \text{ for all } i \in S_\ell.$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}_5^*]_2$ because it does not know $[a]_2$, but the above instances are computable without $[\mathbf{b}_5^*]_2$. For the challenge, $\mathcal{B}$ set $[\mathbf{c}_i]_1$ for all $i \in [m^*]$ in the challenge ciphertext as

$$\begin{aligned}[\mathbf{c}_i]_1 :=& [(\pi_i(1,i), x_i^0, 0, 0, 0, 0)\mathbf{B} + (0, 0, 0, t_\beta, e, 0, 0)\mathbf{W}]_1 \\ =& [(\pi_i(1,i), x_i^0, e, \beta f, 0, 0)\mathbf{B}]_1.\end{aligned}$$

Observe that we can implicitly set $z := e$ and $\tilde{z} := f$, then $\mathcal{A}$'s view is the same as in Game 0 (resp. Game 1) if $\beta = 0$ (resp. $\beta = 1$). □

Lemma 5.20. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_1] - \mathsf{Pr}[\mathsf{E}_2]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$, and sets $\mathsf{pp} := \mathbb{G}$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}, \mathbf{B}^*$ as follows,

$$\mathbf{W} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p),\ \ \mathbf{B} := \begin{pmatrix} \mathbf{I}_3 & & \\ & \begin{matrix} 0 & 1 \\ 1 & -a \end{matrix} & \\ & & \mathbf{I}_2 \end{pmatrix} \mathbf{W},\ \ \mathbf{B}^* := \begin{pmatrix} \mathbf{I}_3 & & \\ & \begin{matrix} a & 1 \\ 1 & 0 \end{matrix} & \\ & & \mathbf{I}_2 \end{pmatrix} \mathbf{W}^* \in \mathsf{GL}_7(\mathbb{Z}_p).$$

Then $\mathcal{B}$ simulates challenge ciphertext as follows.

$$\begin{aligned}\zeta, \eta \leftarrow \mathbb{Z}_p,\ \ [\mathbf{c}_i]_1 :=& [(\pi_i(1,i), x_i^0, 0, 0, 0, 0)\mathbf{B} + (0, 0, 0, \zeta, \eta, 0, 0)\mathbf{W}]_1 \\ =& [(\pi_i(1,i), x_i^0, \eta + a\zeta, \zeta, 0, 0)\mathbf{B}]_1.\end{aligned}$$

Note that $\mathcal{B}$ cannot compute $[\mathbf{b}_5]_1$ because it does not know $[a]_1$, but the above instances are computable without $[\mathbf{b}_5]_1$. Observe that we can implicitly set $z := \eta + a\zeta$ and $\tilde{z} := \zeta$, then $\mathcal{B}$ correctly simulates the challenge ciphertext. For the ℓ-th secret key query for $\ell \in [q_{\mathsf{sk}}]$, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in S_\ell$ as

$$\begin{aligned}& r'_{\ell,i}, r''_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ s.t. } \sum_{i \in S_\ell} r'_{\ell,i} = \sum_{i \in S_\ell} r''_{\ell,i} = 0, \\ & [\mathbf{k}_{\ell,i}]_2 := [(\rho_{\ell,i}(-i, 1), y_{\ell,i}, r'_{\ell,i}, 0, 0, 0)\mathbf{B}^* + r''_{\ell,i}(0, 0, 0, t_\beta, e, 0, 0)\mathbf{W}^*]_2 \\ & = [(\rho_{\ell,i}(-i, 1), y_{\ell,i}, r'_{\ell,i} + e r''_{\ell,i}, \beta f r''_{\ell,i}, 0, 0)\mathbf{B}^*]_2.\end{aligned}$$

We can implicitly set $r_{\ell,i} := r'_{\ell,i} + e r''_{\ell,i}$ and $\tilde{r}_{\ell,i} := f r''_{\ell,i}$ unless $f = 0$, and $\mathcal{A}$'s view is the same as in Game 1 if $\beta = 0$ and Game 2 if $\beta = 1$. □

Lemma 5.21. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_3] = \frac{1}{m_{\mathsf{max}}} \mathsf{Pr}[\mathsf{E}_2].$$

Proof. First, we consider the game (denoted by Game X) that is the same as Game 2 except that $\mathcal{A}$'s output is defined as $\perp$ when $m' \neq m^*$. Note that the challenger does not abort the game in Game X in contrast to Game 3. It is obvious that the probabilities that $\mathcal{A}$ outputs 1 are equal in Game X and Game 3 respectively. Then, we have

$$\begin{aligned}\mathsf{Pr}[\mathsf{E}_3] = \mathsf{Pr}[\mathsf{E}_X] &= \sum_{i\in[m_{\mathsf{max}}]} \mathsf{Pr}[m'=i]\mathsf{Pr}[m^*=i\wedge \mathsf{E}_2|m'=i] \\ &= \frac{1}{m_{\mathsf{max}}}\sum_{i\in[m_{\mathsf{max}}]} \mathsf{Pr}[m^*=i\wedge \mathsf{E}_2] \\ &= \frac{1}{m_{\mathsf{max}}}\mathsf{Pr}[\mathsf{E}_2].\end{aligned}$$

The second line follows from the fact that m' is chosen independently from $\mathcal{A}$'s view in Game X and its value does not affect $\mathcal{A}$'s behavior. □

Lemma 5.22. *Let m_{max} be the maximum length of the challenge vector and s_{max} be the maximum index with which $\mathcal{A}$ queries the key generation oracle. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_3] - \mathsf{Pr}[\mathsf{E}_4]| \leq 4(s_{\mathsf{max}}-1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ that distinguishes the instance of Lemma 5.17 using $\mathcal{A}$. $\mathcal{B}$ first chooses $m' \leftarrow [m_{\mathsf{max}}]$ as a guess of m^*. If the guess is incorrect, i.e., $m' \neq m^*$, then $\mathcal{B}$ outputs 0. Otherwise, $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is. $\mathcal{B}$ obtains an instance of Lemma 5.17 with $n := s_{\mathsf{max}}$ and $m := m'$, and sets $\mathsf{pk} := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1)$. For the challenge, $\mathcal{B}$ sets $[\mathbf{c}_i]_1$ for all $i \in [m^*]$ in the challenge ciphertext as

$$z \leftarrow \mathbb{Z}_p,\ \ [\mathbf{c}_i]_1 := [\mathbf{u}_i + x_i^0\mathbf{b}_3 + z\mathbf{b}_4]_1 = [(\pi_i(1,i), x_i^0, z, \tilde{z}, 0, 0)\mathbf{B}]_1.$$

On the other hand, $\mathcal{B}$ generates the ℓ-th secret key for all ℓ s.t. $(\max S_\ell \leq m') \vee (\min S_\ell > m')$ by using $[\mathbf{b}_1^*]_2, \ldots, [\mathbf{b}_5^*]_2$,

$$\begin{aligned}&\rho_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ s.t. } \sum_{i\in S_\ell} r_{\ell,i} = \sum_{i\in S_\ell}\tilde{r}_{\ell,i} = 0, \\ &[\mathbf{k}_{\ell,i}]_2 := [(\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 \ \text{ for all } i \in S_\ell.\end{aligned}$$

$\mathcal{B}$ generates the ℓ-th secret key for all ℓ s.t. $(\max S_\ell > m') \wedge (\min S_\ell \leq m')$ as

$$\begin{aligned}&\rho_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ for } i \leq m',\ \ \alpha_{\ell,i}, \tilde{\rho}_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ for } i > m', \\ &r_{\ell,i}, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ s.t. } \sum_{i\in S_\ell} r_{\ell,i} = \sum_{i\in S_\ell}\tilde{r}_{\ell,i} = 0, \\ &[\mathbf{k}_{\ell,i}]_2 := \begin{cases} [(\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 & (i \leq m') \\ \begin{aligned}&[\alpha_{\ell,i}\mathbf{u}^*_{i,\beta} + (\tilde{\rho}_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 \\ &=[((\tilde{\rho}_{\ell,i} + \alpha_{\ell,i}\rho'_i)(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} + \beta\alpha_{\ell,i}r'_i, 0, 0)\mathbf{B}^*]_2.\end{aligned} & (i > m') \end{cases}\end{aligned}$$

for all $i \in S_\ell$. Observe that we can set $\rho_{\ell,i} := \tilde{\rho}_{\ell,i} + \alpha_{\ell,i}\rho'_i$ for $i > m'$. We can also set

$$\bar{r}_{\ell,i} := \begin{cases} \tilde{r}_{\ell,i} & (i \in S_\ell, i \le m') \\ \tilde{r}_{\ell,i} + \alpha_{\ell,i} r'_i & (i \in S_\ell, i > m'). \end{cases}$$

unless $r'_i = 0$. This is because the information of $\alpha_{\ell,i}$ in $\rho_{\ell,i}$ is hidden by $\tilde{\rho}_{\ell,i}$, then $\{\tilde{r}_{\ell,i}\}_{i\in S_\ell, i\le m'}$ and $\{\alpha_{\ell,i} r'_i\}_{i\in S_\ell, i>m'}$ are independently random elements in $\mathbb{Z}_p$. Hence, $\mathcal{A}$'s view is the same as in Game 3 if $\beta = 0$ and Game 4 if $\beta = 1$, and we have

$$\begin{aligned} |\mathsf{Pr}[\mathsf{E}_3] - \mathsf{Pr}[\mathsf{E}_4]| &= \left| \sum_{i\in[m_{\mathsf{max}}]} (\mathsf{Pr}[m' = i]\mathsf{Pr}[\mathsf{E}_2 | m' = i] - \mathsf{Pr}[m' = i]\mathsf{Pr}[\mathsf{E}_3 | m' = i]) \right| \\ &\le \sum_{i\in[m_{\mathsf{max}}]} \mathsf{Pr}[m' = i] \times 4(s_{\mathsf{max}} - i)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)} \\ &\le \frac{1}{m_{\mathsf{max}}} \sum_{i\in[m_{\mathsf{max}}]} 4(s_{\mathsf{max}} - 1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)} \\ &= 4(s_{\mathsf{max}} - 1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}. \end{aligned}$$

□

Lemma 5.23. *Let m_{max} be the maximum length of the challenge vector. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_4] - \mathsf{Pr}[\mathsf{E}_5]| \le 8m_{\mathsf{max}}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ that distinguishes the instance of Lemma 5.18 using $\mathcal{A}$. $\mathcal{B}$ first chooses $m' \leftarrow [m_{\mathsf{max}}]$ as a guess of m^*. If the guess is incorrect, i.e., $m' \neq m^*$, then $\mathcal{B}$ outputs 0. Otherwise, $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is. $\mathcal{B}$ obtains an instance of Lemma 5.18 with $n := s_{\mathsf{max}}$ and $m := m'$ and sets $\mathsf{pk} := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1)$. Recall that s_{max} is the maximum index with which $\mathcal{A}$ queries the key generation oracle. For the challenge, $\mathcal{B}$ sets $[\mathbf{c}_i]_1$ for all $i \in [m^*]$ in the challenge ciphertext as

$$\begin{aligned} &z, \tilde{z} \leftarrow \mathbb{Z}_p, \\ &[\mathbf{c}_i]_1 := [\tilde{z}\mathbf{u}_{i,\beta} + (0, 0, x_i^0, z, 0, 0, 0)\mathbf{B}]_1 = [(\tilde{z}\pi'_i(1, i), x_i^0 + \beta w_i \tilde{z}, z, \tilde{z}, 0, 0)\mathbf{B}]_1. \end{aligned}$$

Observe that we can set $\pi_i := \tilde{z}\pi'_i$ unless $\tilde{z} = 0$. On the other hand, for the ℓ-th secret key query s.t. $\max S_\ell \le m'$, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in S_\ell$ as

$$\begin{aligned} &\rho''_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p \ \text{ s.t. } \sum_{i\in S_\ell} r_{\ell,i} = \sum_{i\in S_\ell} \tilde{r}_{\ell,i} = 0, \\ &\mathbf{k}_{\ell,i} := [y_{\ell,i}\mathbf{u}^*_{i,\beta} + (\rho''_{\ell,i}(-i, 1), 0, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 \\ &= [((y_{\ell,i}\rho'_i + \rho''_{\ell,i})(-i, 1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} - \beta w_i y_{\ell,i}, 0, 0)\mathbf{B}^*]_2. \end{aligned}$$

For the ℓ-th secret key query s.t. $(\max S_\ell > m') \wedge (\min S_\ell \leq m')$, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in S_\ell$ as

$$r_{\ell,i}, \bar{r}_{\ell,i}, \rho''_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} r_{\ell,i} = 0,$$

$$\mathbf{k}_{\ell,i} := \begin{cases} [y_{\ell,i}\mathbf{u}^*_{i,\beta} + (\rho''_{\ell,i}(-i,1), 0, r_{\ell,i}, \bar{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 & \\ = [((y_{\ell,i}\rho'_i + \rho''_{\ell,i})(-i,1), y_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i} - \beta w_i y_{\ell,i}, 0, 0)\mathbf{B}^*]_2 & (i \leq m') \\ [y_{\ell,i}\mathbf{u}^*_i + (\rho''_{\ell,i}(-i,1), 0, r_{\ell,i}, \bar{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 & \\ = [((y_{\ell,i}\rho'_i + \rho''_{\ell,i})(-i,1), y_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2. & (i > m') \end{cases}$$

For the ℓ-th secret key query s.t. $\min S_\ell > m'$, $\mathcal{B}$ replies to $\mathcal{A}$ for all $i \in S_\ell$ as

$$\rho''_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} r_{\ell,i} = \sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0,$$

$$\begin{aligned} \mathbf{k}_{\ell,i} &:= [y_{\ell,i}\mathbf{u}^*_i + (\rho''_{\ell,i}(-i,1), 0, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2 \\ &= [((y_{\ell,i}\rho'_i + \rho''_{\ell,i})(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i}, 0, 0)\mathbf{B}^*]_2. \end{aligned}$$

Observe that we can set $\rho_{\ell,i} := y_{\ell,i}\rho'_i + \rho''_{\ell,i}$. Hence, if $m' = m^*$, $\mathcal{A}$'s view is the same as in Game 4 if $\beta = 0$ and Game 5 if $\beta = 1$, and we have

$$\begin{aligned} |\mathsf{Pr}[\mathsf{E}_4] - \mathsf{Pr}[\mathsf{E}_5]| &= \left| \sum_{i \in [m_{\mathsf{max}}]} (\mathsf{Pr}[m' = i]\mathsf{Pr}[\mathsf{E}_4 | m' = i] - \mathsf{Pr}[m' = i]\mathsf{Pr}[\mathsf{E}_5 | m' = i]) \right| \\ &\leq \sum_{i \in [m_{\mathsf{max}}]} \mathsf{Pr}[m' = i] \times 8i\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)} \\ &\leq \frac{1}{m_{\mathsf{max}}} \sum_{i \in [m_{\mathsf{max}}]} 8m_{\mathsf{max}}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)} \\ &= 8m_{\mathsf{max}}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}. \end{aligned}$$

□

Lemma 5.24. *For any PPT adversary $\mathcal{A}$, we have*

$$|\mathsf{Pr}[\mathsf{E}_5] - \mathsf{Pr}[\mathsf{E}_6]| \leq 2^{-\Omega(\lambda)}.$$

Proof. Here, we denote the event such that $m' = m^*$ in Game ι by X_ι. By the game definition, we have

$$\begin{aligned} &|\mathsf{Pr}[\mathsf{E}_5] - \mathsf{Pr}[\mathsf{E}_6]| \\ =&|\mathsf{Pr}[\mathsf{X}_5]\mathsf{Pr}[\mathsf{E}_5|\mathsf{X}_5] - \mathsf{Pr}[\mathsf{X}_6]\mathsf{Pr}[\mathsf{E}_6|\mathsf{X}_6]| \\ =&|\mathsf{Pr}[\mathsf{X}_5](\mathsf{Pr}[\mathsf{E}_5|\mathsf{X}_5] - \mathsf{Pr}[\mathsf{E}_6|\mathsf{X}_6])| \end{aligned}$$

In the third line, we use the fact that $\mathcal{A}$'s view is identical before the ν-th ciphertext query, and we have $\mathsf{Pr}[\mathsf{X}_5] = \mathsf{Pr}[\mathsf{X}_6]$. Therefore, it is sufficient to prove that $|\mathsf{Pr}[\mathsf{E}_5|\mathsf{X}_5] - \mathsf{Pr}[\mathsf{E}_6|\mathsf{X}_6]| \leq 2^{-\Omega(\lambda)}$. For the purpose, we analyze $\mathcal{A}$'s view under the condition such that $m' = m^*$.

We can define $w'_i := w_i - \frac{x_i^1 - x_i^0}{\tilde{z}}$ for $i \in [m^*]$ unless $\tilde{z} = 0$ and observe that w'_i are independently random elements in $\mathbb{Z}_p$. Then we have

$$\mathbf{c}_i = (\pi_i(1,i), x_i^0 + w_i\tilde{z}, z, \tilde{z}, 0, 0)\mathbf{B} = (\pi_i(1,i), x_i^1 + w'_i\tilde{z}, z, \tilde{z}, 0, 0)\mathbf{B}.$$

Next, we check the secret keys. The ℓ-th key for all ℓ s.t. $\max S_\ell \le m^*$ is

$$\begin{aligned}\mathbf{k}_{\ell,i} &= (\rho_{i,\ell}(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} - w_i y_{\ell,i}, 0, 0)\mathbf{B}^* \\ &= \left(\rho_{i,\ell}(-i,1), y_{\ell,i}, r_{\ell,i}, \tilde{r}_{\ell,i} - w'_i y_{\ell,i} - \frac{y_{\ell,i}(x_i^1 - x_i^0)}{\tilde{z}}, 0, 0\right)\mathbf{B}^*.\end{aligned}$$

By the condition Eq. (3.5), we have $\sum_{i\in S_\ell} y_{\ell,i}(x_i^1 - x_i^0) = 0$. Hence we can set $\tilde{r}'_{\ell,i} := \tilde{r}_{\ell,i} - \frac{y_{\ell,i}(x_i^1 - x_i^0)}{\tilde{z}}$. Observe that $\tilde{r}'_{\ell,i}$ is randomly distributed s.t. $\sum_{i\in S_\ell} \tilde{r}'_{\ell,i} = 0$. In the same way, the ℓ-th key for all ℓ s.t. $(\max S_\ell > m^*) \wedge (\min S_\ell \le m^*)$ is

$$\mathbf{k}_{\ell,i} = \begin{cases} \left(\rho_{i,\ell}(-i,1), y_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i} - w'_i y_{\ell,i} - \frac{y_{\ell,i}(x_i^1 - x_i^0)}{\tilde{z}}, 0, 0\right)\mathbf{B}^* & (i \le m^*) \\ (\rho_{i,\ell}(-i,1), y_{\ell,i}, r_{\ell,i}, \bar{r}_{\ell,i}, 0, 0)\mathbf{B}^* & (i > m^*) \end{cases}$$

In this case, we have no condition on $y_{\ell,i}(x_i^1 - x_i^0)$ but $\bar{r}_{\ell,i}$ are independently random elements in $\mathbb{Z}_p$. Then $\bar{r}'_{\ell,i} := \bar{r}_{\ell,i} - \frac{y_{\ell,i}(x_i^1 - x_i^0)}{\tilde{z}}$ for $i \le m^*$ are also independently random elements in $\mathbb{Z}_p$. Finally, the ℓ-th key for all ℓ s.t. $\min S_\ell > m^*$ is not related to this conceptual change. From the above observation, $\mathcal{A}$'s views in Game 5 and Game 6 are identical except the negligible probability. □

Lemma 5.25. *Let m_{max} be the maximum length of the challenge vector and s_{max} be the maximum index with which $\mathcal{A}$ queries the key generation oracle. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_6] - \mathsf{Pr}[\mathsf{E}_7]| \le \{8m_{\mathsf{max}} + 4(s_{\mathsf{max}} - 1)\}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

We can prove Lemma 5.25 by just the reverse of Games 3 to 5.

Lemma 5.26. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_7] = \frac{1}{m_{\mathsf{max}}}\mathsf{Pr}[\mathsf{E}_8].$$

We can prove Lemma 5.26 similarly to Lemma 5.21.

Lemma 5.27. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_8] - \mathsf{Pr}[\mathsf{E}_9]| \le 2\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

We can prove Lemma 5.27 by just the reverse of Games 0 to 2.

5.3.3 Proofs of Lemma 5.17 and Lemma 5.18

We present the proofs that we deferred in the previous section.

Proof of Lemma 5.17. We consider a series of games to prove Lemma 5.17. For each game transition, we show that the difference of probabilities that $\mathcal{A}$ outputs 1 in both games is negligible.

Game 0: This game is the same as the case of $\beta = 0$, i.e., $\mathcal{A}$ is given an instance (D, U_0).

Game 1-μ-1($\mu \in [m+1, n]$): We define Game 0 as Game 1-0-4. This game is the same as Game 1-$(\mu-1)$-4 except that

$$\gamma \leftarrow \mathbb{Z}_p, \quad \mathbf{u}^*_\mu := (\rho'_\mu(-\mu, 1), 0, 0, 0, \boxed{\gamma}, 0)\mathbf{B}^*.$$

That is, $\mathcal{A}$ is given (D, U) where

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \quad \{\pi_i\}_{i\in[m]}, \tilde{z} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}_i := (\pi_i(1, i), 0, 0, \tilde{z}, 0, 0)\mathbf{B} \ \text{ for all } i \in [m],\\
&D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}^*_1]_2, \ldots, [\mathbf{b}^*_5]_2, [\mathbf{u}_1]_1, \ldots, [\mathbf{u}_m]_1),\\
&\{\rho'_i\}_{i\in[m+1,n]}, \{r'_i\}_{i\in[m+1,n]} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}^*_i := \begin{cases} (\rho'_i(-i, 1), 0, 0, r'_i, 0, 0)\mathbf{B}^* & \text{if } i \in [m+1, \mu-1],\\ (\rho'_i(-i, 1), 0, 0, 0, \gamma, 0)\mathbf{B}^* & \text{if } i = \mu,\\ (\rho'_i(-i, 1), 0, 0, 0, 0, 0)\mathbf{B}^* & \text{if } i \in [\mu+1, n], \end{cases}\\
&U := \{[\mathbf{u}^*_i]_2\}_{i\in[m+1,n]}.
\end{aligned}$$

Game 1-μ-2($\mu \in [m+1, n]$): This game is the same as Game 1-μ-1 except that for all $i \in [m]$,

$$\delta_i \leftarrow \mathbb{Z}_p, \quad \mathbf{u}_i := (\pi_i(1, i), 0, 0, \tilde{z}, \boxed{\delta_i}, 0)\mathbf{B}.$$

Game 1-μ-3($\mu \in [m+1, n]$): This game is the same as Game 1-μ-2 except that

$$r'_\mu \leftarrow \mathbb{Z}_p, \quad \mathbf{u}^*_\mu := (\rho'_\mu(-\mu, 1), 0, 0, \boxed{r'_\mu}, \gamma, 0)\mathbf{B}^*.$$

Game 1-μ-4($\mu \in [m+1, n]$): This game is the same as Game 1-μ-3 except that

$$\begin{aligned}
&\mathbf{u}^*_\mu := (\rho'_\mu(-\mu, 1), 0, 0, r'_\mu, \boxed{0}, 0)\mathbf{B}^*,\\
&\mathbf{u}_i := (\pi_i(1, i), 0, 0, \tilde{z}, \boxed{0}, 0)\mathbf{B} \ \text{ for all } i \in [m].
\end{aligned}$$

Observe that Game 1-n-4 is the same as the case of $\beta = 1$, i.e., $\mathcal{A}$ is given an instance (D, U_1).

In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game ι by E_ι,

Lemma 5.28. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{\text{1-}(\mu-1)\text{-4}}] - \mathsf{Pr}[\mathsf{E}_{\text{1-}\mu\text{-1}}]| \le \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}, \mathbf{B}^*$ as follows,

$$\mathbf{W} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p),$$

$$\mathbf{B} := \begin{pmatrix} & -\mu & 1 & & & & \\ & 1 & & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & -a & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}, \quad \mathbf{B}^* := \begin{pmatrix} & & 1 & & & & \\ a & 1 & \mu & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^* \in \mathsf{GL}_7(\mathbb{Z}_p).$$

Observe that $(\mathbf{B}, \mathbf{B}^*)$ are random dual orthonormal bases. $\mathcal{B}$ can compute $[\mathbf{B}]_1$ and $[\mathbf{B}^*]_2$ except $[\mathbf{b}_6]_1$ because $\mathcal{B}$ does not know $[a]_1$. Hence, $\mathcal{B}$ can compute all vectors in D and $\{[\mathbf{u}_i^*]_2\}_{i\in[m+1,n],i\neq\mu}$. Finally, $\mathcal{B}$ computes

$$[\mathbf{u}_\mu^*]_2 := [(t_\beta, e, 0, 0, 0, 0, 0)\mathbf{W}^*]_2 = [(e(-\mu, 1), 0, 0, 0, \beta f, 0)\mathbf{B}^*]_2.$$

We can define $\rho'_\mu := e$ and $\gamma := f$, then if $\beta = 0$ (resp. $\beta = 1$), $\mathcal{A}$ obtains the instance of Game 1-$(\mu-1)$-4 (resp. Game 1-μ-1). □

Lemma 5.29. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}2}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 1$, i.e., $(\mathbb{G}, [a]_1, [e]_1, [t_\beta]_1)$. $\mathcal{B}$ define a matrix $\mathbf{R} \in \mathsf{GL}_7(\mathbb{Z}_p)$ as

$$\mathbf{R} := \begin{pmatrix} 1 & 1 & \\ -1 & -\mu & \\ & & \mathbf{I}_5 \end{pmatrix}, \quad \mathbf{R}^* := \frac{1}{1-\mu} \begin{pmatrix} -\mu & 1 & \\ -1 & 1 & \\ & & (1-\mu)\mathbf{I}_5 \end{pmatrix}.$$

Then $\mathcal{B}$ defines random dual orthonormal bases $(\mathbf{B}, \mathbf{B}^*)$ as follows,

$$\mathbf{W} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p),$$

$$\mathbf{B} := \mathbf{R}^{-1} \begin{pmatrix} a & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ 1 & & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W} = \frac{1}{1-\mu} \begin{pmatrix} -\mu a & -\mu & -1 & & & & \\ a & 1 & 1 & & & & \\ & & & 1-\mu & & & \\ & & & & 1-\mu & & \\ & & & & & 1-\mu & \\ 1-\mu & & & & & & \\ & & & & & & 1-\mu \end{pmatrix} \mathbf{W},$$

$$\mathbf{B}^* := \mathbf{R}^{\top} \begin{pmatrix} & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ 1 & -a & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^* = \begin{pmatrix} 1 & -1 & & & & & \\ 1 & -\mu & & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ 1 & -a & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^*.$$

Observe that $\mathcal{B}$ can compute $[\mathbf{B}]_1, [\mathbf{B}^*]_2$ except $[\mathbf{b}_6^*]_2$. Hence, $\mathcal{B}$ can compute $[\mathbf{u}_i^*]_2$ for all i s.t. $(i \in [m+1, n]) \wedge (i \neq \mu)$. Next $\mathcal{B}$ computes $[\mathbf{u}_\mu^*]_2$ as

$$\zeta, \eta \leftarrow \mathbb{Z}_p,$$

$$[\mathbf{u}_\mu^*]_2 := [(\zeta, \eta, 0, 0, 0, 0, 0, 0)\mathbf{W}^*]_2 = \left[(\eta + a\zeta, 0, 0, 0, 0, 0, \zeta, 0) \begin{pmatrix} & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ 1 & -a & & & & & & \\ & & & & & & & 1 \end{pmatrix} \mathbf{W}^* \right]_2$$

$$= [(\eta + a\zeta, 0, 0, 0, 0, \zeta, 0)\mathbf{R}^*\mathbf{B}^*]_2 = \left[\left(\frac{\eta + a\zeta}{1-\mu}(-\mu, 1), 0, 0, 0, 0, \zeta, 0 \right) \mathbf{B}^* \right]_2.$$

We can implicitly set $\rho'_\mu := \frac{\eta + a\zeta}{1-\mu}$ and $\gamma := \zeta$. Hence, $\mathcal{B}$ can simulate all elements in U. Finally, $\mathcal{B}$ computes $[\mathbf{u}_i]_1$ for all $i \in [m]$ as,

$$\begin{aligned}
&\tilde{z}, \pi'_i, \pi''_i \leftarrow \mathbb{Z}_p, \quad e_i := e\pi'_i + \pi''_i, \\
&[\mathbf{u}_i]_1 := \left[\left(t_\beta \pi'_i + a\pi''_i, e_i, \frac{(i-1)e_i}{i-\mu}, 0, 0, \tilde{z}, 0 \right) \mathbf{W} \right]_1 \\
&= \left[\left(e_i, \frac{(i-1)e_i}{i-\mu}, 0, 0, \tilde{z}, \beta f \pi'_i, 0 \right) \begin{pmatrix} a & 1 & & & & & \\ & & 1 & & 1 & & \\ & & & 1 & & 1 & \\ 1 & & & & & & 1 \end{pmatrix} \mathbf{W} \right]_1 \\
&= \left[\left(e_i, \frac{(i-1)e_i}{i-\mu}, 0, 0, \tilde{z}, \beta f \pi'_i, 0 \right) \mathbf{R}\mathbf{B} \right]_1 \\
&= \left[\left(\frac{(1-\mu)e_i}{i-\mu}(1, i), 0, 0, \tilde{z}, \beta f \pi'_i, 0 \right) \mathbf{B} \right]_1.
\end{aligned}$$

Observe that e_i and π'_i are independent because the information of π'_i in e_i is hidden by π''_i. Then, we can define $\pi_i := \frac{(1-\mu)e_i}{i-\mu}$ and $\delta_i := f\pi'_i$ unless $f = 0$. We can see that if $\beta = 0$ (resp. $\beta = 1$), $\mathcal{A}$ obtains the instance of Game 1-μ-1 (resp. Game 1-μ-2). □

Lemma 5.30. *For any PPT adversary $\mathcal{A}$, we have*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}2}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}3}]| \leq 2^{-\Omega(\lambda)}.$$

Proof. We define $(\mathbf{D}, \mathbf{D}^*)$ as

$$r'_\mu \leftarrow \mathbb{Z}_p, \quad \mathbf{D} := \begin{pmatrix} \mathbf{I}_4 & & & \\ & 1 & \frac{r'_\mu}{\gamma} & \\ & & 1 & \\ & & & 1 \end{pmatrix} \mathbf{B}, \quad \mathbf{D}^* := \begin{pmatrix} \mathbf{I}_4 & & & \\ & 1 & & \\ & -\frac{r'_\mu}{\gamma} & 1 & \\ & & & 1 \end{pmatrix} \mathbf{B}^* \in \mathsf{GL}_7(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}_i^*)$ are random dual orthonormal bases and this basis change does not affect the public vectors, namely, $\mathbf{b}_1, \ldots, \mathbf{b}_4, \mathbf{b}_1^*, \ldots, \mathbf{b}_5^*$. That is, we have

$$\mathbf{d}_i = \begin{cases} \mathbf{b}_i & i \in \{1,2,3,4,6,7\} \\ \mathbf{b}_5 + \frac{r'_\mu}{\gamma}\mathbf{b}_6 & i = 5 \end{cases}$$

$$\mathbf{d}_i^* = \begin{cases} \mathbf{b}_i^* & i \in \{1,2,3,4,5,7\} \\ -\frac{r'_\mu}{\gamma}\mathbf{b}_5^* + \mathbf{b}_6^* & i = 6 \end{cases}$$

Next, we check $\mathbf{u}_i$ for all $i \in [m]$.

$$\begin{aligned} \mathbf{u}_i =& (\pi_i(1,i), 0, 0, \tilde{z}, \delta_i, 0)\mathbf{B} = (\pi_i(1,i), 0, 0, \tilde{z}, \delta_i, 0) \begin{pmatrix} \mathbf{I}_4 & & & \\ & 1 & -\frac{r'_\mu}{\gamma} & \\ & & 1 & \\ & & & 1 \end{pmatrix} \mathbf{D} \\ =& \left(\pi_i(1,i), 0, 0, \tilde{z}, \delta_i - \frac{r'_\mu \tilde{z}}{\gamma}, 0\right) \mathbf{D}. \end{aligned}$$

Observe that we can define $\delta'_i := \delta_i - \frac{r'_\mu \tilde{z}}{\gamma}$, then δ'_i for all $i \in [m]$ are distributed randomly in $\mathbb{Z}_p$. Finally, we check $\mathbf{u}_i^*$ for all $i \in [m+1, n]$.

$$\begin{aligned} \mathbf{u}_i^* =& (\rho'_i(-i,1), 0, 0, \beta_i r'_i, \hat{\beta}_i \gamma, 0)\mathbf{B}^* = (\rho'_i(-i,1), 0, 0, \beta_i r'_i, \hat{\beta}_i \gamma, 0) \begin{pmatrix} \mathbf{I}_4 & & & \\ & 1 & & \\ & \frac{r'_\mu}{\gamma} & 1 & \\ & & & 1 \end{pmatrix} \mathbf{D}^* \\ =& (\rho'_i(-i,1), 0, 0, \beta_i r'_i + \hat{\beta}_i r'_\mu, \hat{\beta}_i \gamma, 0)\mathbf{D}^* \end{aligned}$$

where $\beta_i = 0$ if $i \geq \mu$ and $\beta_i = 1$ if $i < \mu$, and $\hat{\beta}_i = 0$ if $i \neq \mu$ and $\hat{\beta}_i = 1$ if $i = \mu$. From the above observation, the instances that $\mathcal{A}$ obtains are identically distributed in Game 1-μ-2 and Game 1-μ-3 unless $\gamma = 0$. □

Lemma 5.31. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}4}]| \leq 2\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda).$$

This lemma can be proven by just the reverse of Game 1-$(\mu-1)$-4 to Game 1-μ-2, so we omit the proof.

From above lemmas, Lemma 5.17 holds. □

Proof of Lemma 5.18. We consider a series of games to prove Lemma 5.18. For each game transition, we show that the difference of probabilities that $\mathcal{A}$ outputs 1 in both games is negligible.

Game 0: This game is the same as the case of $\beta = 0$, i.e., $\mathcal{A}$ is given an instance (D, U_0).

Game 1-μ-1($\mu \in [m]$): We define Game 0 as Game 1-0-6. This game is the same as Game 1-($\mu-1$)-6 except that

$$\gamma \leftarrow \mathbb{Z}_p, \ \ \mathbf{u}_\mu := (\pi'_\mu(1,\mu), 0, 0, 1, \boxed{\gamma}, 0)\mathbf{B}.$$

That is, $\mathcal{A}$ is given (D, U) where

$$\begin{aligned}
&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \ \ \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \ \ \{\rho'_i\}_{i\in[m+1,n]} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}^*_i := (\rho'_i(-i,1), 1, 0, 0, 0, 0, 0)\mathbf{B}^* \ \text{ for all } i \in [m+1, n],\\
&D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}^*_1]_2, [\mathbf{b}^*_2]_2, [\mathbf{b}^*_4]_2, [\mathbf{b}^*_5]_2, \{[\mathbf{u}^*_i]_2\}_{i\in[m+1,n]}),\\
&\{\pi'_i\}_{i\in[m]}, \{\rho'_i\}_{i\in[m]}, \{w_i\}_{i\in[m]} \leftarrow \mathbb{Z}_p,\\
&\mathbf{u}_i := \begin{cases} (\pi'_i(1,i), w_i, 0, 1, 0, 0)\mathbf{B} & \text{if } i \in [\mu-1],\\ (\pi'_i(1,i), 0, 0, 1, \gamma, 0)\mathbf{B} & \text{if } i = \mu,\\ (\pi'_i(1,i), 0, 0, 1, 0, 0)\mathbf{B} & \text{if } i \in [\mu+1, m], \end{cases}\\
&\mathbf{u}^*_i := \begin{cases} (\rho'_i(-i,1), 1, 0, -w_i, 0, 0)\mathbf{B}^* & \text{if } i \in [\mu-1],\\ (\rho'_i(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* & \text{if } i \in [\mu, m], \end{cases}\\
&U := \{[\mathbf{u}_i]_1, [\mathbf{u}^*_i]_2\}_{i\in[m]}.
\end{aligned}$$

Game 1-μ-2($\mu \in [m]$): This game is the same as Game 1-μ-1 except that for all i s.t. $(i \in [n]) \wedge (i \neq \mu)$,

$$\delta_i \leftarrow \mathbb{Z}_p, \ \ \mathbf{u}^*_i := (\rho'_i(-i,1), 1, 0, -\beta_i w_i, \boxed{\delta_i}, 0)\mathbf{B}^*,$$

where $\beta_i = 0$ if $i > \mu$ and $\beta_i = 1$ if $i < \mu$.

Game 1-μ-3($\mu \in [m]$): This game is the same as Game 1-μ-2 except that,

$$\gamma' \leftarrow \mathbb{Z}_p, \ \ \mathbf{u}^*_\mu := (\rho'_\mu(-\mu,1), 1, 0, 0, 0, \boxed{\gamma'})\mathbf{B}^*.$$

Game 1-μ-4($\mu \in [m]$): This game is the same as Game 1-μ-3 except that for all i s.t. $(i \in [m]) \wedge (i \neq \mu)$,

$$\delta'_i \leftarrow \mathbb{Z}_p, \ \ \mathbf{u}_i := (\pi'_i(1,i), \beta_i w_i, 0, 1, 0, \boxed{\delta'_i})\mathbf{B},$$

where $\beta_i = 0$ if $i > \mu$ and $\beta_i = 1$ if $i < \mu$.

Game 1-μ-5($\mu \in [m]$): This game is the same as Game 1-μ-4 except that,

$$\begin{aligned}
\gamma, \gamma', w_\mu \leftarrow \mathbb{Z}_p, \ \ \mathbf{u}_\mu &:= (\pi'_\mu(1,\mu), \boxed{w_\mu}, 0, 1, \gamma, 0)\mathbf{B},\\
\mathbf{u}^*_\mu &:= (\rho'_\mu(-\mu,1), 1, 0, \boxed{-w_\mu}, 0, \gamma')\mathbf{B}^*.
\end{aligned}$$

Game 1-μ-6($\mu \in [m]$): This game is the same as Game 1-μ-5 except that

$$\begin{aligned}
w_i \leftarrow \mathbb{Z}_p, \ \ \mathbf{u}_i &:= (\pi'_\mu(1,i), \beta_i w_i, 0, 1, \boxed{0,0})\mathbf{B} \ \text{ for all } i \in [m]\\
\mathbf{u}^*_i &:= (\rho'_i(-i,1), 1, 0, -\beta_i w_i, \boxed{0,0})\mathbf{B}^* \ \text{ for all } i \in [n],
\end{aligned}$$

where $\beta_i = 0$ if $i > \mu$ and $\beta_i = 1$ if $i \leq \mu$. Observe that Game 1-m-6 is the same as the case of $\beta = 1$, i.e., $\mathcal{A}$ is given an instance (D, U_1).

In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game ι by E_ι,

Lemma 5.32. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}(\mu-1)\text{-}6}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 1$, i.e., $(\mathbb{G}, [a]_1, [e]_1, [t_\beta]_1)$. $\mathcal{B}$ defines random dual orthonormal bases $\mathbf{B}, \mathbf{B}^*$ as follows,

$$\mathbf{W} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p),$$

$$\mathbf{B} := \begin{pmatrix} a & 1 & -\mu & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}, \quad \mathbf{B}^* := \begin{pmatrix} & 1 & & & & & \\ & \mu & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & -a & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^* \in \mathsf{GL}_7(\mathbb{Z}_p).$$

Observe that $(\mathbf{B}, \mathbf{B}^*)$ are random dual orthonormal bases. $\mathcal{B}$ can compute $[\mathbf{B}]_1$ and $[\mathbf{B}^*]_2$ except $[\mathbf{b}^*_6]_2$ because $\mathcal{B}$ does not know $[a]_2$. Hence, $\mathcal{B}$ can compute all vectors in $D, \{[\mathbf{u}_i]_1\}_{i\in[m],i\neq\mu}$, and $\{[\mathbf{u}^*_i]_2\}_{i\in[m]}$. Finally, $\mathcal{B}$ computes

$$[\mathbf{u}_\mu]_1 := [(t_\beta, e, 0, 0, 0, 0, 0)\mathbf{W}]_1 = [(e(1, \mu), 0, 0, 0, \beta f, 0)\mathbf{B}]_1.$$

We can define $\pi'_\mu := e$ and $\gamma := f$, then if $\beta = 0$ (resp. $\beta = 1$), $\mathcal{A}$ obtains the instance of Game 1-$(\mu-1)$-6 (resp. Game 1-μ-1). □

Lemma 5.33. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}1}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}2}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. We show that we can make a reduction algorithm $\mathcal{B}$ for the SXDH using $\mathcal{A}$. $\mathcal{B}$ obtains an instance of SXDH with $\iota := 2$, i.e., $(\mathbb{G}, [a]_2, [e]_2, [t_\beta]_2)$. $\mathcal{B}$ define a matrix $\mathbf{R} \in \mathsf{GL}_7(\mathbb{Z}_p)$ as

$$\mathbf{R} := \begin{pmatrix} 1 & \mu & \\ -1 & & \\ & & \mathbf{I}_5 \end{pmatrix}, \quad \mathbf{R}^* := \frac{1}{\mu} \begin{pmatrix} & 1 & \\ -\mu & 1 & \\ & & \mu\mathbf{I}_5 \end{pmatrix}.$$

Then $\mathcal{B}$ defines random dual orthonormal bases $(\mathbf{B}, \mathbf{B}^*)$ as follows,

$$\mathbf{W} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p),$$
$$\mathbf{B} := \mathbf{R}^{-1} \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ 1 & -a & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \mathbf{W} = \frac{1}{\mu} \begin{pmatrix} \mu & & & -\mu & & & \\ & 1 & & 1 & & & \\ & & \mu & & & & \\ & & & \mu & & & \\ & & & & \mu & & \\ \mu & -\mu a & & & & \mu & \\ & & & & & & \mu \end{pmatrix} \mathbf{W},$$
$$\mathbf{B}^* := \mathbf{R}^{\top} \begin{pmatrix} a & 1 & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ 1 & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^* = \begin{pmatrix} a & 1 & -1 & & & & \\ & & \mu & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & & & & & & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^*.$$

Observe that $\mathcal{B}$ can compute $[\mathbf{B}]_1, [\mathbf{B}^*]_2$ except $[\mathbf{b}_6]_1$. Then $\mathcal{B}$ can compute $\{[\mathbf{u}_i]_1\}_{i\in[m],i\neq\mu}$ and $[\mathbf{u}^*_\mu]_2$. Next $\mathcal{B}$ computes $[\mathbf{u}_\mu]_1$ as

$$\zeta, \eta \leftarrow \mathbb{Z}_p,$$
$$[\mathbf{u}_\mu]_1 := [(\zeta, \eta, 0, 0, 0, 0, 0, 0)\mathbf{W}]_1 = \left[(\eta + a\zeta, 0, 0, 0, 0, 0, \zeta, 0) \begin{pmatrix} & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ 1 & & -a & & & & & \\ & & & & & & & 1 \end{pmatrix} \mathbf{W} \right]_1$$
$$= [(\eta + a\zeta, 0, 0, 0, 0, 0, \zeta, 0)\mathbf{R}\mathbf{B}]_1 = [((\eta + a\zeta)(1, \mu), 0, 0, 0, 0, 0, \zeta, 0)\, \mathbf{B}]_1 .$$

We can implicitly set $\pi'_\mu := \eta + a\zeta$ and $\gamma := \zeta$. Finally, $\mathcal{B}$ computes $[\mathbf{u}^*_i]_2$ for all i s.t. $(i \in [n]) \wedge (i \neq \mu)$ as,

$$w_i, \sigma_i, \sigma'_i \leftarrow \mathbb{Z}_p, \quad e_i := e\sigma_i + \sigma'_i,$$
$$[\mathbf{u}^*_i]_2 := \left[\left(t_\beta \sigma_i + a\sigma'_i, e_i, \frac{ie_i}{\mu - i}, 1, 0, -\beta_i w_i, 0 \right) \mathbf{W}^* \right]_2$$
$$= \left[\left(e_i, \frac{ie_i}{\mu - i}, 1, 0, -\beta_i w_i, \beta f \sigma_i, 0 \right) \begin{pmatrix} a & 1 & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ 1 & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \mathbf{W}^* \right]_2$$
$$= \left[\left(e_i, \frac{ie_i}{\mu - i}, 1, 0, -\beta_i w_i, \beta f \sigma_i, 0 \right) \mathbf{R}^* \mathbf{B}^* \right]_2$$
$$= \left[\left(\frac{\mu e_i}{\mu - i}(-i, 1), 1, 0, -\beta_i w_i, \beta f \sigma_i, 0 \right) \mathbf{B}^* \right]_2 .$$

Observe that e_i and σ_i are independent because the information of σ_i in e_i is hidden by σ'_i. Then, we can define $\rho'_i := \frac{\mu e_i}{\mu - i}$ and $\delta_i := f\sigma_i$ unless $f = 0$. We can see that if $\beta = 0$ (resp. $\beta = 1$), $\mathcal{A}$ obtains the instance of Game 1-μ-1 (resp. Game 1-μ-2). □

Lemma 5.34. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}2}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).$$

We can prove Lemma 5.34 almost the same way as Lemma 5.32, so we omit the proof.

Lemma 5.35. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}3}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}4}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

We can prove Lemma 5.35 almost the same way as Lemma 5.33, so we omit the proof.

Lemma 5.36. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}4}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}5}] \leq 2^{-\Omega(\lambda)}.$$

Proof. We define $(\mathbf{D}, \mathbf{D}^*)$ as

$$w_\mu \leftarrow \mathbb{Z}_p,$$

$$\mathbf{D} := \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & -\frac{w_\mu}{\gamma'} \\ & & -\frac{w_\mu}{\gamma} & & & 1 & \frac{w_\mu}{\gamma\gamma'} \\ & & & & & & 1 \end{pmatrix} \mathbf{B}, \quad \mathbf{D}^* := \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \frac{w_\mu}{\gamma} \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & \frac{w_\mu}{\gamma'} & -\frac{w_\mu}{\gamma\gamma'} & 1 \end{pmatrix} \mathbf{B}^* \in \mathsf{GL}_7(\mathbb{Z}_p).$$

Observe that $(\mathbf{D}_i, \mathbf{D}_i^*)$ are random dual orthonormal bases and this basis change does not affect the public vectors, namely, $\mathbf{b}_1, \ldots, \mathbf{b}_4, \mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_4^*, \mathbf{b}_5^*$. That is, we have

$$\mathbf{d}_i = \begin{cases} \mathbf{b}_i & i \in \{1,2,3,4,7\} \\ \mathbf{b}_5 - \frac{w_\mu}{\gamma'}\mathbf{b}_7 & i = 5 \\ -\frac{w_\mu}{\gamma}\mathbf{b}_3 + \mathbf{b}_6 + \frac{w_\mu}{\gamma\gamma'}\mathbf{b}_7 & i = 6 \end{cases}$$

$$\mathbf{d}_i^* = \begin{cases} \mathbf{b}_i^* & i \in \{1,2,4,5,6\} \\ \mathbf{b}_3^* + \frac{w_\mu}{\gamma}\mathbf{b}_6^* & i = 3 \\ \frac{w_\mu}{\gamma'}\mathbf{b}_5^* - \frac{w_\mu}{\gamma\gamma'}\mathbf{b}_6^* + \mathbf{b}_7^* & i = 7 \end{cases}$$

Then, we check $\mathbf{u}_i$ for all $i \in [m]$ and $\mathbf{u}_i^*$ for all $i \in [n]$,

$$\begin{aligned} \mathbf{u}_i =& (\pi_i'(1,i), \beta_i w_i, 0, 1, \hat{\beta}_i\gamma, (1-\hat{\beta}_i)\delta_i)\mathbf{B} \\ =& (\pi_i'(1,i), \beta_i w_i, 0, 1, \hat{\beta}_i\gamma, (1-\hat{\beta}_i)\delta_i) \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \frac{w_\mu}{\gamma'} \\ & & \frac{w_\mu}{\gamma} & & & 1 & -\frac{w_\mu}{\gamma\gamma'} \\ & & & & & & 1 \end{pmatrix} \mathbf{D} \\ =& \left(\pi_i'(1,i), \beta_i w_i + \hat{\beta}_i w_\mu, 0, 1, \hat{\beta}_i\gamma, (1-\hat{\beta}_i)\left(\delta_i + \frac{w_\mu}{\gamma'}\right)\right)\mathbf{D}, \end{aligned}$$

$$
\begin{aligned}
\mathbf{u}_i^* =&(\rho_i'(-i,1),1,0,-\beta_i w_i,(1-\hat{\beta}_i)\delta_i',\hat{\beta}_i\gamma')\mathbf{B}^* \\
=&(\rho_i'(-i,1),1,0,-\beta_i w_i,(1-\hat{\beta}_i)\delta_i',\hat{\beta}_i\gamma')
\begin{pmatrix}
1 & & & & & & \\
& 1 & & & & & \\
& & 1 & & & -\frac{w_\mu}{\gamma} & \\
& & & 1 & & & \\
& & & & 1 & & \\
& & & & & 1 & \\
& & & & -\frac{w_\mu}{\gamma'} & \frac{w_\mu}{\gamma\gamma'} & 1
\end{pmatrix}\mathbf{D}^* \\
=&\left(\rho_i'(-i,1),1,0,-\beta_i w_i-\hat{\beta}_i w_\mu,(1-\hat{\beta}_i)\left(\delta_i'-\frac{w_\mu}{\gamma}\right),\hat{\beta}_i\gamma'\right)\mathbf{D}^*,
\end{aligned}
$$

where $\beta_i = 0$ if $i \geq \mu$ and $\beta_i = 1$ if $i < \mu$, and $\hat{\beta}_i = 0$ if $i \neq \mu$ and $\hat{\beta}_i = 1$ if $i = \mu$. Observe that we can define $\tilde{\delta}_i := \delta_i + \frac{w_\mu}{\gamma'}$ and $\tilde{\delta}_i' := \delta_i' - \frac{w_\mu}{\gamma}$ if $\gamma \neq 0$ and $\gamma' \neq 0$, and $\tilde{\delta}_i$ and $\tilde{\delta}_i'$ are independently random elements in $\mathbb{Z}_p$. From the above observation, the instances that $\mathcal{A}$ obtains are identically distributed in Game 1-μ-4 and Game 1-μ-5 except the negligible probability. □

Lemma 5.37. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$
|\mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}5}] - \mathsf{Pr}[\mathsf{E}_{1\text{-}\mu\text{-}6}]| \leq 4\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.
$$

We can prove Lemma 5.37 by just the reverse of Game 1-$(\mu-1)$-1 to Game 1-μ-4. so we omit the proof.

From the above lemmas, Lemma 5.18 holds. □

5.3.4 Semi-Adaptively Secure Scheme for (E:sep, K:sep, D:ct-dom)

We present our semi-adaptively secure scheme for (E:sep, K:sep, D:ct-dom), of which the function class is defined in Definition 5.1. The construction is the same as our E:con scheme (Section 5.3.1) except that $\mathbf{x} := (x_i)_{i\in[m]}$ is replaced by $\mathbf{x} := (x_i)_{i\in U}$, where $U \subseteq [u]$ for any polynomial $u := u(\lambda)$. The correctness holds in the same manner as our E:con scheme. The security statement is somewhat different from that of our E:con scheme as follows.

Theorem 5.4. *Assume that the SXDH assumption holds, then our Pub-UIPFE is semi-adaptively secure. More formally, let u_{max} be the maximum cardinality of challenge index set that $\mathcal{A}$ outputs and s_{max} be the maximum index with which $\mathcal{A}$ queries the key generation oracle, then for any PPT adversary $\mathcal{A}$ and security parameter λ, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.*

$$
\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Pub\text{-}UIPFE}}(\lambda) \leq \{16u_{\mathsf{max}} + 8(s_{\mathsf{max}} - 1) + 4\}\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.
$$

Proof. The idea is the same as the private-key based scheme in the selective E:sep setting. That is, instead of guessing the index set of the challenge ciphertext, the reduction algorithm knows it before simulating secret keys by considering semi-adaptive setting. Then the game sequence after Game 3 is modified from the E:con scheme as follows. Let U^* be the index set of the challenge vector, and $[m^*]$ is changed to U^* in all games other than the games below.

Game 3: This game is the same as Game 2 except that in the ℓ-th secret key query for all ℓ s.t. $(S_\ell \cap U^* \neq \phi) \wedge (S_\ell \backslash U^* \neq \phi)$, $\mathbf{k}_{\ell,i}$ is set as

$$\bar{r}_{\ell,i} \leftarrow \mathbb{Z}_p,$$
$$\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i}}, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell.$$

Game 4: This game is the same as Game 3 except that $\mathbf{c}_i$ and $\mathbf{k}_{\ell,i}$ in the challenge ciphertext and the ℓ-th secret key for all ℓ s.t. $(S_\ell \cap U^* \neq \phi)$ are generated as

$$w_i, \tilde{r}_{\ell,i}, \bar{r}_{\ell,i} \leftarrow \mathbb{Z}_p \text{ s.t. } \sum_{i \in S_\ell} \tilde{r}_{\ell,i} = 0,$$
$$\mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^0 + w_i \tilde{z}}, z, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in U^*,$$
$$\mathbf{k}_{\ell,i} := \begin{cases} (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\tilde{r}_{\ell,i} - w_i y_{\ell,i}}, 0, 0)\mathbf{B}^* & (i \in U^*, S_\ell \subseteq U^*) \\ (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{\bar{r}_{\ell,i} - w_i y_{\ell,i}}, 0, 0)\mathbf{B}^* & (i \in U^*, S_\ell \backslash U^* \neq \phi) \end{cases}.$$

Game 5: This game is the same as Game 4 except that $\mathbf{c}_i$ in the challenge ciphertext is set as

$$w_i, \tilde{z} \leftarrow \mathbb{Z}_p, \quad \mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^1 + w_i \tilde{z}}, z, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in U^*.$$

Game 6: This game is the same as the real security game where the challenge ciphertext is the encryption of $\mathbf{x}^1$ as described in Definition 3.6. That is, the challenge ciphertext for a pair of vectors $(\mathbf{x}^0, \mathbf{x}^1) \in (\mathbb{Z}^{m^*})^2$ is replied as

$$\mathbf{c}_i := (\pi_i(1,i), \boxed{x_i^1}, z, \boxed{0}, 0, 0)\mathbf{B} \text{ for all } i \in U^*$$
$$\mathsf{ct}_{m^*} := ([\mathbf{c}_1]_1, \ldots, [\mathbf{c}_{m^*}]_1).$$

The ℓ-th secret key query with an index set S_ℓ and vector $\mathbf{y}_\ell \in \mathbb{Z}^{S_\ell}$ is replied as

$$\mathbf{k}_{\ell,i} := (\rho_{\ell,i}(-i,1), y_{\ell,i}, r_{\ell,i}, \boxed{0}, 0, 0)\mathbf{B}^* \text{ for all } i \in S_\ell$$
$$\mathsf{sk}_{\ell,S_\ell} := (S_\ell, \{[\mathbf{k}_{\ell,i}]_2\}_{i \in S_\ell}).$$

We modify the lemmas 5.17 and 5.18 to be suitable for the E:sep setting.

Lemma 5.38. *For any polynomial $n := n(\lambda)$ and any set $M \subseteq [m]$ with any polynomial $m := m(\lambda)$, we define the following distribution,*

$$\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \quad \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \quad \{\pi_i\}_{i \in M}, \tilde{z} \leftarrow \mathbb{Z}_p,$$
$$\mathbf{u}_i := (\pi_i(1,i), 0, 0, \tilde{z}, 0, 0)\mathbf{B} \text{ for all } i \in M,$$
$$D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}_1^*]_2, \ldots, [\mathbf{b}_5^*]_2, \{[\mathbf{u}_i]_1\}_{i \in M}),$$
$$\{\rho_i'\}_{i \in [n] \backslash M}, \{r_i'\}_{i \in [n] \backslash M} \leftarrow \mathbb{Z}_p,$$
$$\mathbf{u}_{i,\beta}^* := (\rho_i'(-i,1), 0, 0, \beta r_i', 0, 0)\mathbf{B}^* \text{ for all } i \in [n] \backslash M,$$
$$U_\beta := \{[\mathbf{u}_{i,\beta}^*]_2\}_{i \in [n] \backslash M}.$$

For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.

$$\begin{aligned}\mathsf{Adv}^{\mathsf{P1}}_{\mathcal{A}}(\lambda) :=& |\mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_0)] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_1)]| \\ \leq& 4|[n]\backslash M|\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.\end{aligned}$$

Lemma 5.39. *For any polynomial $n := n(\lambda)$ and any set $M \subseteq [m]$ with any polynomial $m := m(\lambda)$, we define the following distribution,*

$$\begin{aligned}&\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \ \ \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p), \ \ \{\rho'_i\}_{i\in[n]\backslash M} \leftarrow \mathbb{Z}_p, \\ &\mathbf{u}^*_i := (\rho'_i(-i,1),1,0,0,0,0,0)\mathbf{B}^* \ \text{ for all } i \in [n]\backslash M, \\ &D := (\mathbb{G}, [\mathbf{b}_1]_1, \ldots, [\mathbf{b}_4]_1, [\mathbf{b}^*_1]_2, [\mathbf{b}^*_2]_2, [\mathbf{b}^*_4]_2, [\mathbf{b}^*_5]_2, \{[\mathbf{u}^*_i]_2\}_{i\in[n]\backslash M}), \\ &\{\pi'_i\}_{i\in[m]}, \{\rho'_i\}_{i\in[m]}, \{w_i\}_{i\in M} \leftarrow \mathbb{Z}_p, \\ &\mathbf{u}_{i,\beta} := (\pi'_i(1,i), \beta w_i, 0, 1, 0, 0)\mathbf{B} \ \text{ for all } i \in M, \\ &\mathbf{u}^*_{i,\beta} := (\rho'_i(-i,1), 1, 0, -\beta w_i, 0, 0)\mathbf{B}^* \ \text{ for all } i \in M, \\ &U_\beta := \{[\mathbf{u}_{i,\beta}]_1, [\mathbf{u}^*_{i,\beta}]_2\}_{i\in M}.\end{aligned}$$

For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the SXDH s.t.

$$\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{A}}(\lambda) := |\mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_0)] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(D, U_1)]| \leq 8|M|\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

The proofs of Lemma 5.38 and Lemma 5.39 is similar to those of Lemma 5.17 and Lemma 5.18 respectively.

Applying these lemmas, we have

$$\begin{aligned}&|\mathsf{Pr}[\mathsf{E}_2] - \mathsf{Pr}[\mathsf{E}_3]| \leq 4(s_{\mathsf{max}} - 1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}, \\ &|\mathsf{Pr}[\mathsf{E}_3] - \mathsf{Pr}[\mathsf{E}_4]| \leq 8u_{\mathsf{max}}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}. \\ &|\mathsf{Pr}[\mathsf{E}_4] - \mathsf{Pr}[\mathsf{E}_5]| \leq 2^{-\Omega(\lambda)}, \\ &|\mathsf{Pr}[\mathsf{E}_5] - \mathsf{Pr}[\mathsf{E}_6]| \leq \{8u_{\mathsf{max}} + 4(s_{\mathsf{max}} - 1) + 2\}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.\end{aligned}$$

□

5.4 Conclusion of Chapter 5

In Chapter 5, we constructed the first unbounded IPFE schemes. Previously, all known IPFE schemes are bounded, that is, we need to fix a vector length in the setup phase and can use only vectors with the fixed length in the scheme. Our unbounded IPFE scheme enables us to use vectors with any length for encryption and key generation. We provides a function-hiding unbounded IPFE scheme and public-key unbounded IPFE scheme. The former utilizes pseudorandom functions and the latter utilizes the entropy amplification technique to achieve unboundedness. The proof strategy is inspired by that for our efficient function-hiding IPFE scheme in Chapter 4.

Chapter 6

Tightly Secure Inner Product Functional Encryption

In this chapter, we present our contribution in [Tom20], which proposes first tightly secure IPFE schemes including function-hiding and multi-input schemes.

6.1 Technical Overview

6.1.1 Tightly Secure Inner Product Functional Encryption

Our scheme is secure in the multi-user and multi-challenge setting under the MDDH assumption, but here we describe our first scheme based on the DDH assumption in the single-user and multi-challenge setting to ease the exposition. Note that the second scheme is also constructed based on the same idea. Our starting point is the adaptively secure IPFE scheme by Agrawal *et al.* [ALS16]. We briefly describe their scheme below. Let m be a vector length in the scheme.

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

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

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

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

Next, we explain the security proof of this scheme by Abdalla *et al.* [AGRW17], 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 $s' \leftarrow \mathbb{Z}_p$, $\mathbf{b} := (1, 0)^\top$, and $\beta \leftarrow \{0, 1\}$. This change is computationally indistinguishable under the DDH assumption. At this point, we redefine $\mathbf{W}$ as

$$\mathbf{W} := \widetilde{\mathbf{W}} + u(\mathbf{x}^1 - \mathbf{x}^0)\mathbf{a}^{\perp\top}, \tag{6.1}$$

where $u \leftarrow \mathbb{Z}_p$, $\widetilde{\mathbf{W}} \leftarrow \mathbb{Z}_p^{m\times 2}$, and $\mathbf{a}^\perp := (1, -a)^\top$, 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

$$\mathbf{W}\mathbf{a} = \widetilde{\mathbf{W}}\mathbf{a}, \tag{6.2}$$

$$\mathbf{W}^\top \mathbf{y}_{\boldsymbol{\ell}} = \widetilde{\mathbf{W}}^\top \mathbf{y}_{\boldsymbol{\ell}} \quad (\ell \text{ is an index for the query number}), \tag{6.3}$$

$$\begin{aligned} \mathbf{W}(s\mathbf{a} + s'\mathbf{b}) + \mathbf{x}^\beta &= \widetilde{\mathbf{W}}(s\mathbf{a} + s'\mathbf{b}) + us'(\mathbf{x}^1 - \mathbf{x}^0) + \mathbf{x}^\beta \\ &= \widetilde{\mathbf{W}}(s\mathbf{a} + s'\mathbf{b}) + (us' + \beta)(\mathbf{x}^1 - \mathbf{x}^0) + \mathbf{x}^0. \end{aligned} \tag{6.4}$$

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. (6.3) follows from this condition. Finally, from Eq. (6.4), 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 $O(q_{\mathsf{ct}})$, 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. (6.1).

The first attempt to obtain a tight reduction is setting $\mathbf{W}$ in Eq. (6.1) as

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

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 $j \in [q_{\mathsf{ct}}]$, and $\{\mathbf{x}_\iota\}_{\iota\in[L]}$ are a basis of V. In this case, Eq. (6.2) and Eq. (6.3) do not change and Eq. (6.4) 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 $\{[s'_j u_\iota]\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]}$ are indistinguishable from $\{[r_{j,\iota}]\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]}$, which are $q_{\mathsf{ct}}L$ 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 $j \in [q_{\mathsf{ct}}]$, 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 $\{s'_j u_\iota\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]}$ on the exponent forms a synthesizer [NR99], and they are computationally indistinguishable from $q_{\mathsf{ct}}L$ 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

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

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

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

$\mathsf{Enc}(\mathsf{pk}, \mathbf{x})$**:** $\mathbf{s} := (s_1, \ldots, s_m) \leftarrow \mathbb{Z}_p^m, \ \ \mathsf{ct} := ([\mathbf{As}], [\mathbf{WAs} + \mathbf{x}])$.

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

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

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

$$\begin{aligned} &\mathbf{B} := \mathbf{I}_m \otimes (1, 0) \in \mathbb{Z}_p^{2m\times m}, \ \ \mathbf{s}'_j := (s'_{j,1}, \ldots, s'_{j,m}) \leftarrow \mathbb{Z}_p^m, \\ &\mathsf{ct} := ([\mathbf{As}_j + \mathbf{Bs}'_j], [\mathbf{W}(\mathbf{As}_j + \mathbf{Bs}'_j) + \mathbf{x}_j^\beta]). \end{aligned} \tag{6.5}$$

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

$$u \leftarrow \mathbb{Z}_p, \quad \mathbf{W} := \widetilde{\mathbf{W}} + u \sum_{\iota\in[L]} \mathbf{x}_\iota \mathbf{a}_\iota^{\perp\top}, \tag{6.6}$$

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

$$\begin{aligned} &\mathbf{WA} = \widetilde{\mathbf{W}}\mathbf{A}, \\ &\mathbf{W}^\top \mathbf{y}_{\boldsymbol{\ell}} = \widetilde{\mathbf{W}}^\top \mathbf{y}_{\boldsymbol{\ell}}, \\ &\mathbf{W}(\mathbf{As}_j + \mathbf{Bs}'_j) + \mathbf{x}_j^\beta = \widetilde{\mathbf{W}}(\mathbf{As}_j + \mathbf{Bs}'_j) + u \sum_{\iota\in[L]} s'_{j,\iota}\mathbf{x}_\iota + \beta(\mathbf{x}_j^1 - \mathbf{x}_j^0) + \mathbf{x}_j^0. \end{aligned} \tag{6.7}$$

In this case, we can see that $\{[us'_{j,\iota}]\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]} \approx_c \{[r_{j,\iota}]\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]}$, where the left-hand side consists of $q_{\mathsf{ct}}L$ 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 $\mathbf{As}_j$ to $\mathbf{As}_j + \mathbf{Bs}'_j$ as the first step and that from $\{[us'_{j,\iota}]\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]}$ to $\{[r_{j,\iota}]\}_{j\in[q_{\mathsf{ct}}],\iota\in[L]}$ 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. (6.5) as

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

where $\phi(j) := \dim V_j$. Next, we modify the definition of $\mathbf{x}_\iota$ as $\mathbf{x}_\iota := \mathbf{x}^1_{\rho(\iota)} - \mathbf{x}^0_{\rho(\iota)} \in \mathbb{Z}_p^m$ 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. (6.7) 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, $\{[us'_{j,\iota}]\}_{j \in [q_{\mathsf{ct}}], \iota \in [\phi(j)]}$ are computationally indistinguishable from $\{[r_{j,\iota}]\}_{j \in [q_{\mathsf{ct}}], \iota \in [\phi(j)]}$, 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.

6.1.2 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.* [ACF+18] and the function-hiding MIPFE scheme by Datta *et al.* [DOT18]. 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 $\mathsf{msk}'_1, \dots, \mathsf{msk}'_\mu$ (we omit public parameters here). In addition, it chooses μ random vectors $\mathbf{u}_1, \dots, \mathbf{u}_\mu \leftarrow \mathbb{Z}_n^m$ and sets a master secret key of the converted scheme as $\mathsf{msk} := (\mathsf{msk}'_1, \dots, \mathsf{msk}'_\mu, \mathbf{u}_1, \dots, \mathbf{u}_\mu)$. 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 $r_1, \dots, r_\mu \leftarrow \mathbb{Z}_n$ 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 $\mathsf{sk} := (\mathsf{sk}'_1, \dots, \mathsf{sk}'_\mu)$. 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.* [ACF+18], 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

$$\sum_{i\in[\mu]} \langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell} \rangle = \sum_{i\in[\mu]} \langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^1_{i,\ell} \rangle = \sum_{i\in[\mu]} \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^1_{i,\ell} \rangle. \tag{6.8}$$

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 $\tilde{\mathbf{u}}_i \leftarrow \mathbb{Z}_n$. 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. (6.8). 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. (6.8). 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. (6.8)), 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.* [TAO16]. 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. (6.8).

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 $\mathbf{v}_i \leftarrow \mathbb{Z}^m_n$ 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 $\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \leftarrow \mathbb{Z}^m_n$. 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 [ACF+18], 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.

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

In this section, we present two our tightly secure Pub-IPFE schemes (FE for $\mathcal{F}^{\mathsf{IP}}_{m,X,Y}$) and non-function-hiding MIPFE schemes (MIFE for $\mathcal{F}^{\mathsf{MIP}}_{m,\mu,X,Y}$), the latter is obtained by applying the conversion by Abdalla et al. [ACF+18] to our IPFE scheme.

6.2.1 First scheme

Construction. The security of the first scheme is tightly reduced to the $\mathcal{D}_k$-MDDH assumption. 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.

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

$\mathsf{Setup}(1^m, \mathsf{pp})$**:** 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.

$$\mathbf{W} \leftarrow \mathbb{Z}_p^{m\times k(k+1)m}, \quad \mathbf{A} := \overbrace{\begin{pmatrix} \tilde{\mathbf{A}} & & & \\ & \tilde{\mathbf{A}} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{A}} \end{pmatrix}}^{km \text{ matrices}} \in \mathbb{Z}_p^{k(k+1)m\times k^2 m}, \tag{6.9}$$

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

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

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

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y})$**:** 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 $\mathbf{y}$,

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

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

Security. For security, we have the following theorem.

Theorem 6.1. *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*

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

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 chapter, 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 $j \in [q_{\mathsf{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

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

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

$$\mathbf{b} \leftarrow \mathbb{Z}_p^{k+1} \backslash \mathsf{span}(\tilde{\mathbf{A}}), \;\; \mathbf{B} := \overbrace{\begin{pmatrix} \mathbf{b} & & & \\ & \mathbf{b} & & \\ & & \ddots & \\ & & & \mathbf{b} \end{pmatrix}}^{km \text{ vectors}} \in \mathbb{Z}_p^{k(k+1)m \times km}, \tag{6.10}$$

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

Game 2: The reply for ciphertext queries is changed as follows. Let $\rho_i : [\phi_i(q_{\mathsf{ct},i})] \to [q_{\mathsf{ct},i}]$ 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 $j \in [q_{\mathsf{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

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

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 $j \in [q_{\mathsf{ct},i}]$, the j-th ciphertext that $\mathcal{A}$ obtains from the oracle corresponds to

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

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

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

Thanks to Lemma 6.1 to Lemma 6.5, 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 the $\mathcal{D}_k$-MDDH s.t.*

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

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 $n := km q_{\mathsf{ct}}$, where $\delta \in \{0, 1\}$. Note that the number $n = km q_{\mathsf{ct}}$ 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 $\mathbf{W}_1, \ldots, \mathbf{W}_\mu \leftarrow \mathbb{Z}_p^{m \times k(k+1)m}$ 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. (6.9). 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 $\beta \leftarrow \{0,1\}$. 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:

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

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 $\mathbb{Z}_p^{k+1}$. Thus, we can rewrite $\mathbf{c}_{j,1}$ as:

$$\begin{aligned}
&\mathbf{s}_{j,1},\dots,\mathbf{s}_{j,km} \leftarrow \mathbb{Z}_p^k,\ \ s'_{j,1},\dots,s'_{j,k\phi_i(j)} \leftarrow \mathbb{Z}_p,\\
&\mathbf{c}_{j,1} = (\tilde{\mathbf{A}}\mathbf{s}_{j,1}+\delta s'_{j,1}\mathbf{b},\dots,\tilde{\mathbf{A}}\mathbf{s}_{j,k\phi_i(j)}+\delta s'_{j,k\phi_i(j)}\mathbf{b},\tilde{\mathbf{A}}\mathbf{s}_{j,k\phi_i(j)+1},\dots,\tilde{\mathbf{A}}\mathbf{s}_{j,km})\\
&\qquad = \mathbf{A}\mathbf{s}_j + \delta\mathbf{B}\mathbf{s}'_j,
\end{aligned}$$

where $\mathbf{s}_j := (\mathbf{s}_{j,1},\dots,\mathbf{s}_{j,km})$ and $\mathbf{s}'_j := (s_{j,1},\dots,s_{j,k\phi_i(j)},0^{k(m-\phi_i(j))})$. 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 6.1. □

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

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

Proof. Lemma 6.2 follows from Claim 6.1 and Claim 6.2. To prove Lemma 6.2, we use a kind of complexity leveraging argument. In the following, we randomly choose vectors independently from the security game as $\{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathsf{ct},i}]} \leftarrow \mathbb{Z}_p^m$. The purpose is to assure that $\tilde{\mathbf{x}}_j$ is independent from $\widetilde{\mathbf{W}}_i$ in Eq. (6.12). □

Claim 6.1. *For any PPT adversary $\mathcal{A}$ and both $\iota \in \{1,2\}$, we have*

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

Proof. Vectors $\{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathsf{ct},i}]}$ 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 6.2. *For any PPT adversary $\mathcal{A}$, we have*

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

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. (6.10). 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:

$$\mathbf{b}_\iota^{*\top}\mathbf{A} = \mathbf{0}^\top,\quad \mathbf{b}_\iota^{*\top}\mathbf{b}_{\iota'} = \begin{cases} 1 & (\iota=\iota') \\ 0 & (\iota\neq\iota') \end{cases} \quad \text{for all } \iota,\iota' \in [km]. \tag{6.11}$$

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

$$\begin{aligned}&\mathbf{u} \leftarrow \mathbb{Z}_p^k,\ \ \widetilde{\mathbf{W}}_i \leftarrow \mathbb{Z}_p^{m\times k(k+1)m},\\ &\mathbf{W}_i := \widetilde{\mathbf{W}}_i + \sum_{\iota\in[\phi_i(q_{\mathsf{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top \left(\mathbf{b}^*_{k(\iota-1)+1}||\dots||\mathbf{b}^*_{k(\iota-1)+k}\right)^\top.\end{aligned} \tag{6.12}$$

Observe that $\mathbf{W}_i$ is identically distributed to the original one, i.e., $\mathbf{W}_i \leftarrow \mathbb{Z}_p^{m\times k(k+1)m}$. 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)

$$\mathbf{W}_i\mathbf{A} = \widetilde{\mathbf{W}}_i\mathbf{A} \quad \text{for all } i\in[\mu]\ , \tag{6.13}$$

(In the secret keys)

$$\mathbf{W}_i^\top \mathbf{y}_\ell = \widetilde{\mathbf{W}}_i^\top \mathbf{y}_\ell \quad \text{for all } i\in[\mu] \text{ and } \ell\in[q_{\mathsf{sk},i}], \tag{6.14}$$

(In the challenge ciphertexts)

$$\begin{aligned}\mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) &= \left(\widetilde{\mathbf{W}}_i + \sum_{\iota\in[\phi_i(q_{\mathsf{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top \left(\mathbf{b}^*_{k(\iota-1)+1}||\dots||\mathbf{b}^*_{k(\iota-1)+k}\right)^\top\right)(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j)\\ &= \widetilde{\mathbf{W}}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \sum_{\iota\in[\phi_i(q_{\mathsf{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top \left(\mathbf{O}_{k\times k(\iota-1)}||\mathbf{I}_k||\mathbf{O}_{k\times k(m-\iota)}\right)\mathbf{s}'_j\\ &= \widetilde{\mathbf{W}}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \sum_{\iota\in[\phi_i(j)]} \langle \mathbf{u}, \tilde{\mathbf{s}}_{j,\iota}\rangle \tilde{\mathbf{x}}_{\rho_i(\iota)} \quad \text{for all } i\in[\mu] \text{ and } j\in[q_{\mathsf{ct},i}].\end{aligned} \tag{6.15}$$

Here, Eq. (6.13) and Eq. (6.15) follow from Eq. (6.11), and Eq. (6.14) follows from Eq. (3.1). Then, from Eq. (6.13), Eq. (6.14), and Eq. (6.15), $\mathcal{A}$'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 6.2 holds. □

Lemma 6.3. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_2$ for the $\mathcal{D}_k$-MDDH s.t.*

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

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

Proof. First, we prove the following claim.

Claim 6.3. *We consider the following distribution for any $n\in\mathbb{N}$:*

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

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

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

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

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

$\mathcal{B}$ chooses $\mathbf{r} \leftarrow \mathbb{Z}_p^k$, 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 6.3. □

We describe a PPT adversary $\mathcal{B}_2$ that solves a problem defined in Claim 6.3 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 $\beta \leftarrow \{0,1\}$. 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:

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

where $\mathbf{A}$ and $\mathbf{B}$ are defined as Eq. (6.9) and Eq. (6.10). 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 6.3. □

Lemma 6.4. *For any PPT adversary $\mathcal{A}$, we have*

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

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

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

Lemma 6.5 is trivial because $\mathcal{A}$ does not obtain any information about β in Game 4.

6.2.2 Second Scheme

Construction. The security of the second scheme is reduced to the $\mathcal{D}_k$-MDDH assumption with a security loss of $O(m^2)$, whereas the numbers of group elements in ciphertexts and secret keys are smaller than the first one. Let 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.

$$\mathbb{G}_{\mathsf{CG}} \leftarrow \mathcal{G}_{\mathsf{CG}}(1^\lambda), \quad \mathbf{A} \leftarrow \mathbb{Z}_p^{k(m+1)\times k}, \quad \mathsf{pp} := (\mathbb{G}_{\mathsf{CG}}, [\mathbf{A}])$$

$\mathsf{Setup}(1^m, \mathsf{pp})$**:** 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.

$$\mathbf{W} \leftarrow \mathbb{Z}_p^{m\times k(m+1)}, \quad \mathsf{pk} := (\mathbb{G}_{\mathsf{CG}}, [\mathbf{A}], [\mathbf{WA}]), \quad \mathsf{msk} := \mathbf{W}.$$

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

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

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

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

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk})$**:** 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 $||\mathbf{x}||_\infty \leq X_\lambda$ and $||\mathbf{y}||_\infty \leq Y_\lambda$, the output of the decryption algorithm is $d = \langle \mathbf{x}, \mathbf{y} \rangle$.

Security. For security, we have the following theorem.

Theorem 6.2. *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 exists a PPT adversary $\mathcal{B}$ for the $\mathcal{D}_k$-MDDH and we have*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{Pub\text{-}IPFE}}_{\mathcal{A}}(\lambda) \leq (km(m+1)+2)\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{CG}}(\lambda) + 2^{-\Omega(\lambda)},\\ &\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),\end{aligned}$$

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

Proof. We employ a series of games and evaluate the advantage of the adversary in each game.

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

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

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

$$\begin{aligned}&\mathbf{B} \leftarrow \mathbb{Z}_p^{k(m+1)\times km} \text{ s.t. } (\mathbf{A}||\mathbf{B}) \text{ is invertible},\\ &\tilde{\mathbf{s}}_{j,1},\ldots,\tilde{\mathbf{s}}_{j,\phi_i(j)} \leftarrow \mathbb{Z}_p^k,\ \ \mathbf{s}'_j := (\tilde{\mathbf{s}}_{j,1},\ldots,\tilde{\mathbf{s}}_{j,\phi_i(j)}, 0^{k(m-\phi_i(j))}) \in \mathbb{Z}_p^{km},\\ &\mathbf{c}_{j,1} := \mathbf{A}\mathbf{s}_j + \boxed{\mathbf{B}\mathbf{s}'_j},\ \ \mathbf{c}_{j,2} := \mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \boxed{\mathbf{B}\mathbf{s}'_j}) + \mathbf{x}_j^\beta.\end{aligned}$$

Games 2 to 4 are defined the same as those in the security proof of the first scheme (Section 6.2.1). Thanks to Lemma 6.6 to Lemma 6.11, Theorem 6.2 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 6.6. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the $\mathcal{D}_k$-MDDH s.t.*

$$\begin{aligned}&|\mathsf{Pr}[\mathsf{E}_0] - \mathsf{Pr}[\mathsf{E}_1]| \leq \frac{km(m+1)}{2}\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{CG}}(\lambda) + 2^{-\Omega(\lambda)},\\ &\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + (\mu + q_{\mathsf{ct}} + q_{\mathsf{sk}})\mathsf{poly}(\lambda, m),\end{aligned}$$

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

To prove Lemma 6.6, we use the following lemma.

Lemma 6.7. *Let $q_1, \ldots, q_m \in \mathbb{N}$ be any natural numbers and $q = \sum_{i\in[m]} q_i$. Consider the following distribution.*

$$\begin{aligned}
&\mathbb{G}_{\mathsf{CG}} \leftarrow \mathcal{G}_{\mathsf{CG}}(1^\lambda),\ \ \mathbf{M} \leftarrow \mathsf{GL}_{k(m+1)}(\mathbb{Z}_p),\ \ \widetilde{\mathbf{S}}_0 \leftarrow \mathbb{Z}_p^{k\times q},\ \ \widetilde{\mathbf{S}}_i \leftarrow \mathbb{Z}_p^{k(i+1)\times q_i},\\
&\mathbf{S}_0 := \begin{pmatrix} \widetilde{\mathbf{S}}_0 \\ \mathbf{O}_{km\times q}\end{pmatrix} \in \mathbb{Z}_p^{k(m+1)\times q},\ \ \mathbf{S}'_i := \begin{pmatrix} \widetilde{\mathbf{S}}_i \\ \mathbf{O}_{k(m-i)\times q_i}\end{pmatrix} \in \mathbb{Z}_p^{k(m+1)\times q_i},\\
&\mathbf{S}_1 := (\mathbf{S}'_1||\cdots||\mathbf{S}'_m) \in \mathbb{Z}_p^{k(m+1)\times q},\\
&\mathbf{T}_0 := \mathbf{M}\mathbf{S}_0,\ \ \mathbf{T}_1 := \mathbf{M}\mathbf{S}_1.
\end{aligned}$$

Then, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the $\mathcal{D}_k$-MDDH s.t.

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{lemma}}_{\mathcal{A}}(\lambda) &:= |\mathsf{Pr}[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{T}_0])] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{T}_1])]|\\
&\le \frac{km(m+1)}{2}\mathsf{Adv}^{\mathcal{D}_k\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{CG}}(\lambda) + 2^{-\Omega(\lambda)}\\
\mathsf{Time}(\mathcal{B}) &\approx \mathsf{Time}(\mathcal{A}) + q\mathsf{poly}(\lambda, m),
\end{aligned}$$

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

Proof of Lemma 6.7. We consider a series of hybrid games, Game j for $j \in \{0, \ldots, m\}$. In Game j, $\mathcal{A}$ is given $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{H}_j])$ where

$$\begin{aligned}
&\mathbb{G}_{\mathsf{CG}} \leftarrow \mathcal{G}_{\mathsf{CG}}(1^\lambda),\ \ \mathbf{M} \leftarrow \mathsf{GL}_{k(m+1)}(\mathbb{Z}_p),\\
&\widetilde{\mathbf{S}}_i \leftarrow \mathbb{Z}_p^{k(i+1)\times q_i} \ \text{ for } i \in [j],\ \ \widetilde{\mathbf{S}}_i \leftarrow \mathbb{Z}_p^{k\times q_i} \ \text{ for } i \in [m]\backslash[j],\\
&\mathbf{S}'_i := \begin{pmatrix} \widetilde{\mathbf{S}}_i \\ \mathbf{O}_{k(m-i)\times q_i}\end{pmatrix} \text{ for } i \in [j],\ \ \mathbf{S}'_i := \begin{pmatrix} \widetilde{\mathbf{S}}_i \\ \mathbf{O}_{km\times q_i}\end{pmatrix} \text{ for } i \in [m]\backslash[j],\\
&\mathbf{H}'_i := \mathbf{M}\mathbf{S}'_i,\ \ \mathbf{H}_j := (\mathbf{H}'_1||\cdots||\mathbf{H}'_m).
\end{aligned}$$

It is easy to see that $\mathbf{H}_0 = \mathbf{T}_0$ and $\mathbf{H}_m = \mathbf{T}_1$. Thus, it is sufficient for the proof to show that $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{H}_{j-1}])$ and $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{H}_j])$ are computationally indistinguishable for $j \in [m]$ under the MDDH assumption. More precisely, we show that, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that

$$|\mathsf{Pr}[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{H}_{j-1}])] - \mathsf{Pr}[1 \leftarrow \mathcal{A}(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{H}_j])]| \le \mathsf{Adv}^{\mathsf{q_j}\text{-}\mathcal{U}_{k(j+1),k}\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{CG}}(\lambda) + 2^{-\Omega(\lambda)}.$$

$\mathcal{B}$ works as follows:

1. $\mathcal{B}$ obtains a q_j-fold $\mathcal{U}_{k(j+1),k}$-MDDH instance $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{A}], [\mathbf{T}_\beta])$.

2. $\mathcal{B}$ defines

$$\mathbf{V} \leftarrow \mathsf{GL}_{k(m+1)}(\mathbb{Z}_p),\ \ \mathbf{B} := \begin{pmatrix} \overline{\mathbf{A}} & \mathbf{O}_{k\times kj} \\ \underline{\mathbf{A}} & \mathbf{I}_{kj} \end{pmatrix} \in \mathbb{Z}_p^{k(j+1)\times k(j+1)},$$
$$\mathbf{M} := \mathbf{V}\begin{pmatrix} \mathbf{B} & \mathbf{O}_{k(j+1)\times k(m-j)} \\ \mathbf{O}_{k(m-j)\times k(j+1)} & \mathbf{I}_{k(m-j)} \end{pmatrix} \in \mathbb{Z}_p^{k(m+1)\times k(m+1)},$$
$$\widetilde{\mathbf{S}}_i \leftarrow \mathbb{Z}_p^{k(i+1)\times q_i} \text{ for } i \in [j-1],\ \ \widetilde{\mathbf{S}}_i \leftarrow \mathbb{Z}_p^{k\times q_i} \text{ for } i \in [m]\backslash[j],$$
$$\mathbf{S}'_i := \begin{pmatrix} \widetilde{\mathbf{S}}_i \\ \mathbf{O}_{k(m-i)\times q_i} \end{pmatrix} \text{ for } i \in [j-1],\ \ \mathbf{S}'_i := \begin{pmatrix} \widetilde{\mathbf{S}}_i \\ \mathbf{O}_{km\times q_i} \end{pmatrix} \text{ for } i \in [m]\backslash[j],$$
$$\mathbf{H}'_i := \mathbf{M}\mathbf{S}'_i \text{ for } i \in [m]\backslash j,\ \ \mathbf{H}'_j := \mathbf{V}\begin{pmatrix} \mathbf{T}_\beta \\ \mathbf{O}_{k(m-j)\times q_j} \end{pmatrix},\ \ \mathbf{H} := (\mathbf{H}'_1||\cdots||\mathbf{H}'_m)$$

where $\overline{\mathbf{A}}$ is the matrix consisting of the first k rows of $\mathbf{A}$, and $\underline{\mathbf{A}}$ is that consisting of the last kj rows of $\mathbf{A}$.

3. $\mathcal{B}$ gives $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{H}])$ to $\mathcal{A}$ and outputs $\mathcal{A}$'s output as it is.

Since $\mathbf{B}$ is invertible with overwhelming probability, $\mathbf{M}$ is distributed statistically close to uniform in $\mathsf{GL}_{k(m+1)}(\mathbb{Z}_p)$. Observe that $\mathbf{H}$ is identically distributed to $\mathbf{H}_{j-1}$ if $\beta = 0$ and is to $\mathbf{H}_j$ if $\beta = 1$. We remark that due to the random self-reducibility of the MDDH assumption, we have

$$\mathsf{Adv}^{\mathsf{q_j}\text{-}\mathcal{U}_{(j+1)\mathsf{k},\mathsf{k}}\text{-}\mathsf{MDDH}}_{\mathcal{B},\mathsf{CG}}(\lambda) \le kj\mathsf{Adv}^{\mathcal{D}_\mathsf{k}\text{-}\mathsf{MDDH}}_{\mathcal{B}',\mathsf{CG}}(\lambda).$$

This concludes the proof. □

Proof of Lemma 6.6. We prove that $\forall\mathcal{A}\exists\mathcal{B}, |\mathsf{Pr}[\mathsf{E}_0] - \mathsf{Pr}[\mathsf{E}_1]| \le \mathsf{Adv}^{\mathsf{lemma}}_{\mathcal{B}}(\lambda)$. Let $q_\iota := |\{(i,j) \mid \phi_i(j) = \iota\}|$ for $\iota \in [m]$. $\mathcal{B}$ works as follows. $\mathcal{B}$ takes an instance of Lemma 6.7, $(\mathbb{G}_{\mathsf{CG}}, [\mathbf{M}], [\mathbf{T}_\delta])$, where $\delta \in \{0,1\}$. $\mathcal{B}$ defines $\mathbf{A}$ as the matrix consisting of the first k columns of $\mathbf{M}$. $\mathcal{B}$ generates random matrices $\mathbf{W}_1, \ldots, \mathbf{W}_\mu \leftarrow \mathbb{Z}_p^{m\times k(m+1)}$ and sets $\mathsf{pk}_i := (\mathbb{G}_{\mathsf{CG}}, [\mathbf{A}], [\mathbf{W}_i\mathbf{A}])$ for all $i \in [\mu]$. Then, $\mathcal{B}$ inputs $\{\mathsf{pk}_i\}_{i\in[\mu]}$ to $\mathcal{A}$. Because $\mathcal{B}$ 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, $\mathcal{B}$ selects a bit $\beta \leftarrow \{0,1\}$. Let $\mathbf{T}_{\delta,i}$ be the matrix consisting of the $(\sum_{j\in[i-1]} q_j + 1)$ to $(\sum_{j\in[i-1]} q_j + q_i)$-th columns. Let $\mathbf{t}_{\delta,i,j}$ be the j-th column of $\mathbf{T}_{\delta,i}$. 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, $\mathcal{B}$ computes a reply as follows:

$$\mathbf{c}_{j,1} := \mathbf{t}_{\delta,\phi_i(j),\sigma_{i,j}(i,j)},\ \ \mathbf{c}_{j,2} := \mathbf{W}_i\mathbf{c}_{j,1} + \mathbf{x}_j^\beta \in \mathbb{Z}_p^m,$$
$$\mathsf{ct}_j := ([\mathbf{c}_{j,1}], [\mathbf{c}_{j,2}]).$$

where $\sigma_{i,j} : \{(\ell,\kappa) \mid \phi_\ell(\kappa) = \phi_i(j)\} \to [q_{\phi_i(j)}]$ be any bijective function.

We check that $\mathcal{B}$ correctly simulates $\mathcal{O}_{\mathsf{ct}}$. We can define $(\mathbf{A}||\mathbf{B}) := \mathbf{M}$ since $(\mathbf{A}||\mathbf{B})$ is invertible with overwhelming probability. From the definition of Lemma 6.7, $\mathbf{t}_{\delta,\phi_i(j),\sigma_{i,j}(i,j)}$ is uniformly distributed in the space spanned by columns of $\mathbf{A}$ if $\delta = 0$, and it is in the space spanned by the first to $k(\phi_i(j)+1)$-th columns of $\mathbf{M}$ if $\delta = 1$. Hence, if $\delta = 0$, $\mathcal{A}$'s view corresponds to Game 0 and otherwise, it corresponds to Game 1. Finally, $\mathcal{B}$ outputs the truth value of $(\beta = \beta')$ where β' is the output of $\mathcal{A}$. This proves Lemma 6.6. □

Lemma 6.8. *For any PPT adversary $\mathcal{A}$, we have*

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

Proof. Lemma 6.8 follows from Claim 6.4 and Claim 6.5. To prove Lemma 6.8, we use a kind of complexity leveraging argument. In the following, we randomly choose vectors independently from the security game as $\{\tilde{\mathbf{x}}_j\}_{i\in[\mu],j\in[q_{\mathrm{ct},i}]} \leftarrow \mathbb{Z}_p^m$. The purpose is to assure that $\tilde{\mathbf{x}}_j$ is independent from $\widetilde{\mathbf{W}}_i$ in Eq. (6.17). □

Claim 6.4. *For any PPT adversary $\mathcal{A}$ and both $\iota \in \{1,2\}$, we have*

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

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

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

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

$$\mathbf{b}_\iota^{*\top}\mathbf{A} = \mathbf{0}^\top, \quad \mathbf{b}_\iota^{*\top}\mathbf{b}_{\iota'} = \begin{cases} 1 & (\iota = \iota') \\ 0 & (\iota \neq \iota') \end{cases} \quad \text{for all } \iota,\iota' \in [km]. \tag{6.16}$$

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

$$\begin{aligned}&\mathbf{u} \leftarrow \mathbb{Z}_p^k, \ \ \widetilde{\mathbf{W}}_i \leftarrow \mathbb{Z}_p^{m\times k(m+1)},\\ &\mathbf{W}_i := \widetilde{\mathbf{W}}_i + \sum_{\iota\in[\phi_i(q_{\mathrm{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top \left(\mathbf{b}^*_{k(\iota-1)+1}||\dots||\mathbf{b}^*_{k(\iota-1)+k}\right)^\top.\end{aligned} \tag{6.17}$$

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

(mod p), we have

(In the public key)

$$\mathbf{W}_i\mathbf{A} = \widetilde{\mathbf{W}}_i\mathbf{A} \quad \text{for all } i \in [\mu] \ , \tag{6.18}$$

(In the secret keys)

$$\mathbf{W}_i^\top \mathbf{y}_\ell = \widetilde{\mathbf{W}}_i^\top \mathbf{y}_\ell \quad \text{for all } i \in [\mu] \text{ and } \ell \in [q_{\mathsf{sk},i}], \tag{6.19}$$

(In the challenge ciphertexts)

$$\begin{aligned}
\mathbf{W}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) &= \left(\widetilde{\mathbf{W}}_i + \sum_{\iota \in [\phi_i(q_{\mathsf{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top \left(\mathbf{b}^*_{k(\iota-1)+1} || \dots || \mathbf{b}^*_{k(\iota-1)+k}\right)^\top\right)(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) \\
&= \widetilde{\mathbf{W}}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \sum_{\iota \in [\phi_i(q_{\mathsf{ct},i})]} \tilde{\mathbf{x}}_{\rho_i(\iota)}\mathbf{u}^\top \left(\mathbf{O}_{k\times k(\iota-1)} || \mathbf{I}_k || \mathbf{O}_{k\times k(m-\iota)}\right)\mathbf{s}'_j \qquad (6.20) \\
&= \widetilde{\mathbf{W}}_i(\mathbf{A}\mathbf{s}_j + \mathbf{B}\mathbf{s}'_j) + \sum_{\iota \in [\phi_i(j)]} \langle \mathbf{u}, \tilde{\mathbf{s}}_{j,\iota}\rangle \tilde{\mathbf{x}}_{\rho_i(\iota)} \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}].
\end{aligned}$$

Here, Eq. (6.18) and Eq. (6.20) follow from Eq. (6.16), and Eq. (6.19) follows from Eq. (3.1). Then, from Eq. (6.18), Eq. (6.19), and Eq. (6.20), $\mathcal{A}$'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 6.5 holds. □

Lemma 6.9. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the $\mathcal{D}_k$-MDDH s.t.*

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

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

Lemma 6.10. *For any PPT adversary $\mathcal{A}$, we have*

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

Lemma 6.11. *For any PPT adversary $\mathcal{A}$, we have*

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

The proofs of Lemma 6.9 to Lemma 6.11 are almost the same as those of Lemma 6.3 to Lemma 6.5, respectively.

6.2.3 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. [ACF+18] 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 $\mathsf{Pub\text{-}IPFE} := (\mathsf{Par}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ be a Pub-IPFE scheme. In their conversion, we require that $\mathsf{Pub\text{-}IPFE}$ has the following properties.

1. $\mathsf{Pub\text{-}IPFE}$ 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 $|z| \leq mX_\lambda Y_\lambda$, we have
$$\mathsf{Pr}\left[\begin{array}{l|l} d = E(\langle \mathbf{x}, \mathbf{y} \rangle \mod n) & \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda) \\ (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^m, \mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{x}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbf{y}) \\ d := \mathsf{Dec}_1(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}) \end{array} \end{array}\right] = 1, \quad \mathsf{Dec}_2(\mathsf{pp}, E(z)) = z.$$
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. [ACF+18]. 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 $\mathsf{Priv\text{-}IPFE}$.

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

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

$\mathsf{Dec}(\mathsf{pp}, \{\mathsf{ct}_i\}_{i\in[\mu]}, \mathsf{sk})$**:** 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 6.1. *Let* MIPFE *be the MIPFE scheme obtained by applying the conversion in [ACF+18] 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*

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

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

6.3 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 [Lin17]. We can apply her framework to our schemes and obtain a tightly function-hiding IPFE scheme in the multi-user setting. Although we consider to apply her framework to the first scheme in this section, we can similarly apply it to the second scheme. 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 $\mathbf{c}_{\mathsf{in}} := (\mathbf{c}_{\mathsf{in},1}, \mathbf{c}_{\mathsf{in},2}) := (\mathbf{As}, \mathbf{WAs} + \mathbf{x}) \in \mathbb{Z}_p^{(k^2+k+1)m}$ 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 6.2.1. 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 $\mathsf{ct} := (\mathbf{k}_{\mathsf{out},1}, \mathbf{k}_{\mathsf{out},2}) := (-\mathbf{V}^\top\mathbf{c}_{\mathsf{in}} \in \mathbb{Z}_p^{k(k+1)(k^2+k+1)m}, \mathbf{c}_{\mathsf{in}} \in \mathbb{Z}_p^{(k^2+k+1)m})$. 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.

6.3.1 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 $\mathbf{k}_{\mathsf{out},\ell} := \mathbf{k}_{\mathsf{out},\ell}^1 - \mathbf{k}_{\mathsf{out},\ell}^0 := (-\mathbf{W}^\top \mathbf{y}_\ell^1, \mathbf{y}_\ell^1) - (-\mathbf{W}^\top \mathbf{y}_\ell^0, \mathbf{y}_\ell^0)$ 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.

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.

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

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

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

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

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

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

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

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

$$\mathbf{c}_{\mathsf{out},1} := \mathbf{Dr}, \quad \mathbf{c}_{\mathsf{out},2} := \mathbf{VDr} + \mathbf{k}_{\mathsf{in}}, \quad \mathsf{sk} := ([\mathbf{c}_{\mathsf{out},1}]_2, [\mathbf{c}_{\mathsf{out},2}]_2).$$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk})$**:** 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 $||\mathbf{x}||_\infty \le X_\lambda$ and $||\mathbf{y}||_\infty \le Y_\lambda$, the output of the decryption algorithm is $d = \langle \mathbf{x}, \mathbf{y} \rangle$.

Security. For security, we have the following theorem.

Theorem 6.3. *Assume that the $\mathcal{D}_k$-MDDH assumption holds with respect to $\mathcal{G}_{\mathsf{BG}}$, 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*

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

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

Theorem 6.3 follows from Theorem 6.1 and Lin's observation [Lin17]. That is, the following relations hold:

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

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 $\langle \mathbf{x}^0_{i,j_i}, \mathbf{y}^0_{i,\ell_i} \rangle = \langle \mathbf{x}^1_{i,j_i}, \mathbf{y}^0_{i,\ell_i} \rangle$ for the inner scheme and $\langle \mathbf{c}^1_{\mathsf{in},i,j_i}, \mathbf{k}^0_{\mathsf{in},i,\ell_i} \rangle = \langle \mathbf{c}^1_{\mathsf{in},i,j_i}, \mathbf{k}^1_{\mathsf{in},i,\ell_i} \rangle$ 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 6.1. 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 [LV16]. 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

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

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

6.4 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 [ACF+18], 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.

6.4.1 Conversion

Property. Let $\mathsf{Priv\text{-}IPFE} := (\mathsf{Par}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ be a Priv-IPFE scheme. 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.
$$\mathsf{pp}' \leftarrow \mathsf{Par}'(1^\lambda), \;\; \{\mathsf{msk}'_i\}_{i\in[\mu]} \leftarrow \mathsf{Setup}'(1^{2m+1}, \mathsf{pp}'), \;\; \{\mathbf{u}_i\}_{i\in[\mu]} \xleftarrow{\cup} \mathbb{Z}_n^m,$$
$$\mathsf{pp} := \mathsf{pp}', \;\; \mathsf{msk} := (\{\mathsf{msk}'_i\}_{i\in[\mu]}, \{\mathbf{u}_i\}_{i\in[\mu]}).$$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, i, \mathbf{x})$**:** 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.

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

$\mathsf{Dec}(\mathsf{pp}, \{\mathsf{ct}_i\}_{i\in[\mu]}, \mathsf{sk})$**:** 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 6.2. Typically, we define Priv-IPFE as consisting of four algorithms ($\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec}$) 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 [ACF+18]. Thus, the security reduction will not become tight.

6.4.2 Security

Theorem 6.4. *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 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 $\mathsf{Time}(\mathcal{A})$.

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 6.1: 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.

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 6.1. A formal description of each game is given as follows. Similarly to in Section 6.2.1, 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

$$\{r_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i \rangle \right) \in \mathbb{Z}_n, \tag{6.21}$$

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

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:

$$\boxed{\{r'_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r'_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]} \left(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i \rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i \rangle \right) \right) \in \mathbb{Z}_n,} \tag{6.22}$$

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

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:

$$\boxed{\{r''_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r''_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r''_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle\right) \in \mathbb{Z}_n,} \tag{6.23}$$

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

Thanks to Lemma 6.12 to Lemma 6.17 and the observation in Section 6.4.2, Theorem 6.4 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 6.12. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_1$ for Priv-IPFE s.t.*

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

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, 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. (6.21). 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 6.13. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_2$ for* Priv-IPFE *s.t.*

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

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, chosen by the game for weakly function-hiding Priv-IPFE. $\mathcal{B}_2$ behaves as follows.

1. $\mathcal{B}_2$ 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. $\mathcal{B}_2$ 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

$$\{r_{i,\ell}\}_{i\in[\mu-1]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n, \quad r_{\mu,\ell} := -\left(\sum_{i\in[\mu-1]} r_{i,\ell} + \sum_{i\in[\mu]} \langle \mathbf{y}^\beta_{i,\ell}, \mathbf{u}_i\rangle\right) \in \mathbb{Z}_n,$$
$$r'_{i,\ell} := r_{i,\ell} - \langle \mathbf{y}^0_{i,\ell}, \mathbf{v}_i\rangle, \quad \tilde{\mathbf{y}}^0_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, 0^m, r_{i,\ell}) \in \mathbb{Z}_n^{2m+1}, \quad \tilde{\mathbf{y}}^1_{i,\ell} := (\mathbf{y}^\beta_{i,\ell}, \mathbf{y}^0_{i,\ell}, r'_{i,\ell}) \in \mathbb{Z}_n^{2m+1}$$
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 6.14. *For any PPT adversary $\mathcal{A}$, we have*

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

Proof. Lemma 6.14 follows from Claim 6.6 and Claim 6.7. To prove Lemma 6.14, we use a kind of complexity leveraging argument. In the following, we randomly choose vectors independently from the security game as $\{\hat{\mathbf{x}}^{\gamma}_{i,1}\}_{\gamma\in\{0,1\},i\in[\mu]} \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m$. 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 6.7. □

Claim 6.6. *For any PPT adversary $\mathcal{A}$ and both $\iota \in \{2,3\}$, we have*

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

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

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

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 $\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m$ for all $i \in [\mu]$. Observe that $\mathbf{u}_i$ and $\mathbf{v}_i$ are identically distributed to the original ones, i.e., $\mathbf{u}_i, \mathbf{v}_i \xleftarrow{\mathsf{U}} \mathbb{Z}_n^m$. 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)

$$\begin{aligned}r'_{\mu,\ell} &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^{\beta}_{i,\ell}, \mathbf{u}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \mathbf{v}_i\rangle\right)\right)\\ &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^{\beta}_{i,\ell}, \tilde{\mathbf{u}}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \tilde{\mathbf{v}}_i\rangle\right) + \sum_{i\in[\mu]}\left(-\langle \mathbf{y}^{\beta}_{i,\ell}, \hat{\mathbf{x}}^{\beta}_{i,1}\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \hat{\mathbf{x}}^{0}_{i,1}\rangle\right)\right) \qquad (6.24)\\ &= -\left(\sum_{i\in[\mu-1]} r'_{i,\ell} + \sum_{i\in[\mu]}\left(\langle \mathbf{y}^{\beta}_{i,\ell}, \tilde{\mathbf{u}}_i\rangle + \langle \mathbf{y}^{0}_{i,\ell}, \tilde{\mathbf{v}}_i\rangle\right)\right) \quad \text{for all } \ell \in [q_{\mathsf{sk}}],\end{aligned}$$

(In the ciphertexts)

$$\tilde{\mathbf{x}}_{i,j} = (\mathbf{x}^{\beta}_{i,j} + \mathbf{u}_i, \mathbf{v}_i, 1) = (\mathbf{x}^{\beta}_{i,j} - \hat{\mathbf{x}}^{\beta}_{i,1} + \tilde{\mathbf{u}}_i, \hat{\mathbf{x}}^{0}_{i,1} + \tilde{\mathbf{v}}_i, 1) \quad \text{for all } i \in [\mu] \text{ and } j \in [q_{\mathsf{ct},i}]. \qquad (6.25)$$

Eq. (6.24) 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. (6.24) and Eq. (6.25), $\mathcal{A}$'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 6.7. □

Lemma 6.15. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_3$ for Priv-IPFE s.t.*

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

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

Proof. We use the following claim in the proof of Lemma 6.15.

Claim 6.8. *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

$$\langle \mathbf{x}^{\beta}_{i,j}, \mathbf{y}^{\beta}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{\beta}_{\iota,1}, \mathbf{y}^{\beta}_{\iota,\ell} \rangle = \langle \mathbf{x}^{0}_{i,j}, \mathbf{y}^{0}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{0}_{\iota,1}, \mathbf{y}^{0}_{\iota,\ell} \rangle \tag{6.26}$$

$$\langle \mathbf{x}^{\beta}_{i,1}, \mathbf{y}^{\beta}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{\beta}_{\iota,1}, \mathbf{y}^{\beta}_{\iota,\ell} \rangle = \langle \mathbf{x}^{0}_{i,1}, \mathbf{y}^{0}_{i,\ell} \rangle + \sum_{\substack{\iota \in [\mu], \\ \iota \neq i}} \langle \mathbf{x}^{0}_{\iota,1}, \mathbf{y}^{0}_{\iota,\ell} \rangle \tag{6.27}$$

Then Eq. (6.26) $-$ Eq. (6.27) yields Claim 6.8. □

Next, we describe $\mathcal{B}_3$'s behavior. Let $\delta \in \{0,1\}$ be a random coin that corresponds to β, 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 $\mathcal{O}_{\mathsf{ct}}$ on $(i, (\tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{x}}^1_{i,j}))$ and obtains $\mathsf{ct}'_{i,j}$ from it. Finally, $\mathcal{B}_3$ 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. (6.22). Then, $\mathcal{B}_3$ 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, $\mathcal{B}_3$ 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 6.8. 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 6.16. *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}_4$ for Priv-IPFE s.t.*

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

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, chosen by the game for weakly function-hiding Priv-IPFE. $\mathcal{B}_4$ behaves as follows.

1. $\mathcal{B}_4$ 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. $\mathcal{B}_4$ 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 $\mathcal{O}_{\mathsf{ct}}$ on $(i, (\tilde{\mathbf{x}}^0_{i,j}, \tilde{\mathbf{x}}^1_{i,j}))$ and obtains $\mathsf{ct}'_{i,j}$ from it. Finally, $\mathcal{B}_4$ 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

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

   Then, $\mathcal{B}_4$ queries $\mathcal{O}_{\mathsf{sk}}$ 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, $\mathcal{B}_4$ 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 6.17. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Pr}[\mathsf{E}_5] = 1/2.$$

Lemma 6.17 is trivial because $\mathcal{A}$ does not obtain any information about β in Game 5.

Optimization

On Lemma 6.12 and Lemma 6.13, we define that

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

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 $i \in [\mu]$, $j \in [q_{\mathsf{ct},i}]$, and $\ell \in [q_{\mathsf{sk}}]$, which satisfies the condition Eq. (3.3). Hence, we do not need Game 1 actually and can prove that

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

Similarly, we can also prove that

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

6.4.3 Application to Our Scheme

Applying the conversion to our scheme presented in Section 6.3.1, we can obtain a tightly secure fully function-hiding MIPFE scheme. First, we confirm that our scheme satisfies the property presented in Section 6.4.1.

1. Theorem 6.3 says that our scheme is weakly function-hiding.
2. We can define that $n := p$, $G := G_T$, and $E : a \in \mathbb{Z}_p \to [a]_T \in G_T$. 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 $g^a_T \cdot g^b_T = g^{a+b}_T$.

Then, from Theorem 6.3 and Theorem 6.4, we obtain the following corollary.

Corollary 6.2. *Let MIPFE be the MIPFE scheme obtained by applying the conversion in Section 6.4.1 to our weakly function-hiding Priv-IPFE scheme. Then MIPFE is fully function-hiding. More formally, let μ be a number of slots, $q_{\mathsf{ct}} := \sum_{i\in[\mu]} q_{\mathsf{ct},i}$ 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 $\mathcal{B}_1, \ldots, \mathcal{B}_4$ for the $\mathcal{D}_k$-MDDH and we have*

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

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

6.5 Conclusion of Chapter 6

In Chapter 6, we studied the tight security of IPFE. Tight security guarantees that the security of the scheme is not degraded even when an adversary obtains many ciphertexts. Since adversaries often obtain many ciphertexts in the real world, tight security is theoretically and practically important. In this work, we first constructed public-key tightly secure IPFE scheme. Using the previous conversion techniques [Lin17,ACF+18], we can convert them into a tightly secure function-hiding scheme and multi-input scheme. To achieve a tightly secure function-hiding multi-input scheme, we devised a new conversion that transforms function-hiding IPFE into function-hiding MIPFE. Applying the conversion to our tightly secure function-hiding IPFE scheme, we finally obtained a tightly secure function-hiding MIPFE scheme.

Chapter 7

Conclusion

7.1 Summary of This Thesis

One of essential roles of cryptography is to hide data from adversaries, and traditional encryption such as public-key encryption (PKE) and symmetric-key encryption (SKE) suffice for the purpose. Due to recent increase in cloud services and interest to privacy issues, demand for computation over encrypted data has been increasing. Unfortunately, PKE and SKE do not allow us to make a computation over encrypted data.

Functional encryption (FE) is an advanced cryptographic paradigm, which allows us to compute function values from encrypted data. Thus, it is expected to be helpful for delegating computation to cloud servers. A problem in this application of FE is that all FE schemes that can handle general functions are too heavy to implement and impractical. Inner Product Functional Encryption (IPFE), introduced to deal with the impracticality, is FE that only supports inner product as a function class. Inner Product Functional Encryption can be efficiently constructed from standard assumptions and has several applications. This is why IPFE has received attention. In this thesis, we proposed methodologies to enhance the efficiency, functionality, and security of IPFE towards the goal of making IPFE more practical.

In Chapter 4, we proposed a more efficient function-hiding IPFE scheme than previous schemes. The function-hiding property is important when computing sensitive functions over encrypted data. To achieve an efficient scheme, we developed a new security proof technique. Our technique has been widely applied to various function-hiding schemes [DOT18,TT18,Tom19]. Concretely, we used an extended proof technique to construct an efficient unbounded IPFE schemes in Chapter 5, and the technique also inspired us to achieve a tightly secure function-hiding IPFE scheme in Chapter 6. Furthermore, the asymptotic efficiency (in vector length) of our scheme in the standard model is still the best. Thus, we consider that our security proof technique is fundamental to achieve efficient function-hiding IPFE schemes.

In Chapter 5, we proposed public-key and private-key unbounded IPFE schemes, which have no bound on vector lengths. We believe that unbounded IPFE schemes are very useful when data to be encrypted have various sizes, e.g., DNA sequences. Since inner product allows us to compute

similarity of two sequences, unbounded IPFE is useful for similarity check over encrypted DNA sequences, which may help to solve some privacy issues in genome research. We also believe that there are many other cases where unboundedness of IPFE make a profit.

In Chapter 6, we developed a methodology to construct tightly secure IPFE schemes. Tight security guarantees that the probability that an adversary breaks a cryptosystem is independent of the number of ciphertexts that the adversary obtains, which is not the case in general. Tight security is practically important because it is natural that an adversary obtains many ciphertexts in the real world. Although it had been unclear whether we can construct tightly secure IPFE schemes, we solved the question affirmatively. We believe that our result makes IPFE more practical because we achieved IPFE schemes that are secure against the realistic threat.

While our technique to construct efficient function-hiding IPFE scheme could be widely used, our results on unboundedness and tight security are incompatible. However, both properties are important towards the goal of obtaining a more practical IPFE scheme. We could not obtain such a scheme in this thesis, and thus this is an important future direction.

We also remark that inner product is a quite simple function class and insufficient for many cases. For instance, when we make a statistical computation, we can compute weighted mean but cannot compute standard deviation via inner product. When we compute similarity of two strings, we can compute the Hamming distance but cannot compute the Levenshtein distance via inner product. Thus, it is also important to explore efficient functional encryption that supports a richer function class than inner product. We hope that our results will help to study such richer FE schemes.

7.2 Other Open Questions

We exhibit other open questions related IPFE in the last part of this thesis.

We proposed unbounded IPFE schemes and tightly secure IPFE schemes based on the MDDH assumptions. A natural question is whether we can construct them from other assumptions such as LWE or DCR, from which (not unbounded or not tightly secure) IPFE schemes have already been constructed. Our techniques are not straightforwardly applicable to LWE or DCR, and thus devising new techniques will need to be devised. They are important because an LWE-based scheme is post-quantum secure, and a DCR-base scheme can handle exponentially large values.

The next question is whether we can construct function-hiding IPFE schemes without pairings. This is important since such a scheme is more efficient than the scheme that uses pairings. So far, no function-hiding schemes have been proposed, and an impossibility result is also known for function-hiding IPFE schemes from LWE [Ünа20]. If new function-hiding IPFE schemes are proposed, then our conversion from function-hiding IPFE to MIPFE may be applicable. Thus, solving this question may lead to new instantiations for function-hiding MIPFE schemes.

Finally, considering unboundedness or tight security in quadratic FE is another interesting open question. Since constructions of quadratic FE are rather different from those of IPFE, we probably need new techniques to achieve these properties in quadratic FE.

Acknowledgements

I deeply thank to Masayuki Abe, a supervisor in my master's course and an examiner of my Ph.D, for giving me insightful comments on the thesis. He also made a great effort to arrange my Ph.D examination. I cannot thank him enough for what he has done for me.

I would like to thank Takayuki Kanda, Yoshimasa Nakamura, Shin-ichi Minato, and Masatoshi Yoshikawa for taking the role of examiner of my Ph.D and giving me helpful feedback on the thesis.

I am grateful to Tatsuaki Okamoto, a supervisor in my master's course. When I applied to his crypto laboratory for taking master's degree, he willingly accepted me, who had been working as a local official for three years after graduating faculty of veterinary medicine! After I joined NTT, he continued to take care of me. I could never be in the current place without him.

I thank to group members in NTT and co-authors for having exiting discussions and helping my study. The group members also supported me to complete the thesis.

Lastly, I thank to Shiori, my wife, for kindly supporting and encouraging me for a long time.

Bibliography

[AAB+15] Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. On the practical security of inner product functional encryption. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 777–798. Springer, Heidelberg, March / April 2015.

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

[ABG19] Michel Abdalla, Fabrice Benhamouda, and Romain Gay. From single-input to multi-client inner-product functional encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 552–582. Springer, Heidelberg, December 2019.

[ABKW19] Michel Abdalla, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. Decentralizing inner-product functional encryption. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 128–157. Springer, Heidelberg, April 2019.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 657–677. Springer, Heidelberg, August 2015.

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

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

[AGVW13] Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 500–518. Springer, Heidelberg, August 2013.

[AHN+17] Masayuki Abe, Dennis Hofheinz, Ryo Nishimaki, Miyako Ohkubo, and Jiaxin Pan. Compact structure-preserving signatures with almost tight security. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 548–580. Springer, Heidelberg, August 2017.

[AHY15] Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. A framework for identity-based encryption with almost tight security. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 521–549. Springer, Heidelberg, November / December 2015.

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

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation with constant size overhead. Cryptology ePrint Archive, Report 2015/1023, 2015. http://eprint.iacr.org/2015/1023.

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

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

[AMY19] Shweta Agrawal, Monosij Maitra, and Shota Yamada. Attribute based encryption (and more) for nondeterministic finite automata from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 765–797. Springer, Heidelberg, August 2019.

[AS16] Prabhanjan Vijendra Ananth and Amit Sahai. Functional encryption for turing machines. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 125–153. Springer, Heidelberg, January 2016.

[Att14] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q.

Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.

[Att16] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, December 2016.

[BBL17] Fabrice Benhamouda, Florian Bourse, and Helger Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 36–66. Springer, Heidelberg, March 2017.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

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

[BCM+15] Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon. Subgroup security in pairing-based cryptography. In Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors, *LATINCRYPT 2015*, volume 9230 of *LNCS*, pages 245–265. Springer, Heidelberg, August 2015.

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, Heidelberg, February 2014.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[BGJS15] Saikrishna Badrinarayanan, Divya Gupta, Abhishek Jain, and Amit Sahai. Multi-input functional encryption for unbounded arity functions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 27–51. Springer, Heidelberg, November / December 2015.

[BJK15] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 470–491. Springer, Heidelberg, November / December 2015.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 1–12. Springer, Heidelberg, August 1998.

[BLR+15] Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 563–594. Springer, Heidelberg, April 2015.

[Bon98] Dan Boneh. The decision diffie-hellman problem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998.

[BR05] John Black and Phillip Rogaway. CBC MACs for arbitrary-length messages: The three-key constructions. *Journal of Cryptology*, 18(2):111–131, April 2005.

[BRS13a] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 461–478. Springer, Heidelberg, August 2013.

[BRS13b] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 255–275. Springer, Heidelberg, December 2013.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 306–324. Springer, Heidelberg, March 2015.

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

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

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 363–384. Springer, Heidelberg, August 2016.

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

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 247–266. Springer, Heidelberg, August 2015.

[CGKW18] Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. Unbounded ABE via bilinear entropy expansion, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 503–534. Springer, Heidelberg, April / May 2018.

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

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Heidelberg, April 2015.

[CLT18] Guilhem Castagnos, Fabien Laguillaumie, and Ida Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 733–764. Springer, Heidelberg, December 2018.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, Heidelberg, December 2001.

[CW13] Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Heidelberg, August 2013.

[CW14] Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, September 2014.

[DDM16] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional encryption for inner product with full function privacy. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 164–195. Springer, Heidelberg, March 2016.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[DOT18] Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 245–277. Springer, Heidelberg, March 2018.

[DP19] Edouard Dufour Sans and David Pointcheval. Unbounded inner-product functional encryption with succinct keys. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 426–441. Springer, Heidelberg, June 2019.

[EHK+17] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, January 2017.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[EM14] Andreas Enge and Jérôme Milan. Implementing cryptographic pairings at standard security levels. In Rajat Subhra Chakraborty, Vashek Matyas, and Patrick 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.

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

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

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 578–602. Springer, Heidelberg, May 2014.

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

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

[GHK17] Romain Gay, Dennis Hofheinz, and Lisa Kohl. Kurosawa-desmedt meets tight security. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 133–160. Springer, Heidelberg, August 2017.

[GHKW16] Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Heidelberg, August 2013.

[GKW18] Romain Gay, Lucas Kowalczyk, and Hoeteck Wee. Tight adaptively secure broadcast encryption with short ciphertexts and keys. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 123–139. Springer, Heidelberg, September 2018.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

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

[GS16] Sanjam Garg and Akshayaram Srinivasan. Single-key to multi-key functional encryption with polynomial loss. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 419–442. Springer, Heidelberg, October / November 2016.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, August 2015.

[HJ12] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, August 2012.

[Hof16] Dennis Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 251–281. Springer, Heidelberg, January 2016.

[Hof17] Dennis Hofheinz. Adaptive partitioning. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 489–518. Springer, Heidelberg, April / May 2017.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995.

[IPS15] Yuval Ishai, Omkant Pandey, and Amit Sahai. Public-coin differing-inputs obfuscation and its applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 668–697. Springer, Heidelberg, March 2015.

[KKS17] Sungwook Kim, Jinsu Kim, and Jae Hong Seo. A new approach for practical function-private inner product encryption. Cryptology ePrint Archive, Report 2017/004, 2017. http://eprint.iacr.org/2017/004.

[KLM+18] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 544–562. Springer, Heidelberg, September 2018.

[KNT18] Fuyuki Kitagawa, Ryo Nishimaki, and Keisuke Tanaka. Obfustopia built on secret-key functional encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 603–648. Springer, Heidelberg, April / May 2018.

[KNTY19] Fuyuki Kitagawa, Ryo Nishimaki, Keisuke Tanaka, and Takashi Yamakawa. Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. Springer, Heidelberg, August 2019.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008.

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

[LM16] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 443–468. Springer, Heidelberg, October / November 2016.

[LPJY15] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, November / December 2015.

[LT19] Benoît Libert and Radu Titiu. Multi-client functional encryption for linear functions in the standard model from LWE. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 520–551. Springer, Heidelberg, December 2019.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In Irit Dinur, editor, *57th FOCS*, pages 11–20. IEEE Computer Society Press, October 2016.

[LW11] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011.

[NR99] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/2010/556.

[OSW07] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 195–203. ACM Press, October 2007.

[OT09] Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231. Springer, Heidelberg, December 2009.

[OT10] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Heidelberg, August 2010.

[OT12a] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 591–608. Springer, Heidelberg, April 2012.

[OT12b] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, December 2012.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

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

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

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

[TAO16] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In Matt Bishop and Anderson

C. A. Nascimento, editors, *ISC 2016*, volume 9866 of *LNCS*, pages 408–425. Springer, Heidelberg, September 2016.

[TAO20] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient inner product functional encryption with full-hiding security. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 103-A(1):33–40, 2020.

[Tom19] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 459–488. Springer, Heidelberg, December 2019.

[Tom20] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. *Theor. Comput. Sci.*, 833:56–86, 2020.

[TT18] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 609–639. Springer, Heidelberg, December 2018.

[TT20] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. *Japan Journal of Industrial and Applied Mathematics*, 37(3):723–779, 2020.

[Ünal20] Akin Ünal. Impossibility results for lattice-based functional encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 169–199. Springer, Heidelberg, May 2020.

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

[Wat12] Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Heidelberg, August 2012.

[Wat15] Brent Waters. A punctured programming approach to adaptively secure functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 678–697. Springer, Heidelberg, August 2015.

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

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

List of Publications Related to the Thesis

Journal Papers

[Tom20] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. *Theor. Comput. Sci.*, 833:56–86, 2020.

[TT20] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. *Japan Journal of Industrial and Applied Mathematics*, 37(3):723–779, 2020.

[TAO20] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient inner product functional encryption with full-hiding security. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 103-A(1):33–40, 2020.

Conference Papers

[Tom19] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 459–488. Springer, Heidelberg, December 2019.

[TT18] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 609–639. Springer, Heidelberg, December 2018.

[TAO16] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In Matt Bishop and Anderson C. A. Nascimento, editors, *ISC 2016*, volume 9866 of *LNCS*, pages 408–425. Springer, Heidelberg, September 2016.

List of All Publications

Journal Papers

[Tom20] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. *Theor. Comput. Sci.*, 833:56–86, 2020.

[TT20] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. *Japan Journal of Industrial and Applied Mathematics*, 37(3):723–779, 2020.

[TAO20] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient inner product functional encryption with full-hiding security. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 103-A(1):33–40, 2020.

Conference Papers

[AT20] Nuttapong Attrapadung and Junichi Tomida. Unbounded dynamic predicate compositions in ABE from standard assumptions. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 405–436. Springer, Heidelberg, December 2020.

[TKN20] Junichi Tomida, Yuto Kawahara, and Ryo Nishimaki. Fast, compact, and expressive attribute-based encryption. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 3–33. Springer, Heidelberg, May 2020.

[Tom19] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 459–488. Springer, Heidelberg, December 2019.

[TFNS19] Junichi Tomida, Atsushi Fujioka, Akira Nagai, and Koutarou Suzuki. Strongly secure identity-based key exchange with single pairing operation. In Kazue Sako, Steve Schnei-

der, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 484–503. Springer, Heidelberg, September 2019.

[TT18] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 609–639. Springer, Heidelberg, December 2018.

[DOT18] Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 245–277. Springer, Heidelberg, March 2018.

[TAO16] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In Matt Bishop and Anderson C. A. Nascimento, editors, *ISC 2016*, volume 9866 of *LNCS*, pages 408–425. Springer, Heidelberg, September 2016.