

# Attribute-Based Access Control for Inner Product Functional Encryption from LWE

Tapas Pal and Ratna Dutta

Department of Mathematics, Indian Institute of Technology Kharagpur,  
Kharagpur-721302, India

`tapas.pal@iitkgp.ac.in`, `ratna@maths.iitkgp.ac.in`

**Abstract.** The notion of functional encryption (FE) was proposed as a generalization of plain public-key encryption to enable a much more fine-grained handling of encrypted data, with advanced applications such as cloud computing, multi-party computations, obfuscating circuits or Turin machines. While FE for general circuits or Turin machines gives a natural instantiation of the many cryptographic primitives, existing FE schemes are based on indistinguishability obfuscation or multilinear maps which either rely on new computational hardness assumptions or heuristically claimed to be secure. In this work, we present new techniques directly yielding FE for inner product functionality where secret-keys provide access control via polynomial-size bounded-depth circuits. More specifically, we encrypt messages with respect to attributes and embed policy circuits into secret-keys so that a restricted class of receivers would be able to learn certain property about the messages. Recently, many inner product FE schemes were proposed. However, none of them uses a general circuit as an access structure. Our main contribution is designing the *first* construction for an attribute-based FE scheme in *key-policy setting* for inner products from well-studied Learning With Errors (LWE) assumption. Our construction takes inspiration from the attribute-based encryption of Boneh et al. from Eurocrypt 2014 and the inner product functional encryption of Agrawal et al. from Crypto 2016. The scheme is proved in a *stronger setting* where the adversary is allowed to ask secret-keys that can decrypt the challenge ciphertext. Doing so requires a careful setting of parameters for handling the noise in ciphertexts to enable correct decryption. Another main advantage of our scheme is that the size of ciphertexts and secret-keys *depends on the depth of the circuits* rather than its size. Additionally, we extend our construction in a much desirable *multi-input* variant where secret-keys are associated with multiple policies subject to different encryption slots. This enhances the applicability of the scheme with finer access control.

**Keywords:** functional encryption, attribute-based encryption, inner product functional encryptions.

## 1 Introduction

Controlling access to encrypted data is an essential requirement in today’s world of cloud computing and data privacy. Plain public-key encryption either hides the entire data or reveals nothing depending on the availability of the secret-key. In many applications of cloud computing, such all-or-nothing type encryption is insufficient. For example, we often need to embed a decryption policy into the secret-key so that only users who satisfy the policy can decrypt the ciphertext. In another scenario, we may want to issue a secret-key that can only let a user learn a specific statistical property of the encrypted data such as average or weighted sum. The notion of (key-policy) *attribute-based encryption* (ABE), introduced by [34,26], is a solution to the former example and the latter can be resolved using *inner product functional encryption* (IPFE) [1] which is a particular class of functional encryption [15]. We consider more general situation where a decryption key requires to serve the functionality of both ABE and IPFE.

To illustrate the potential of the proposed scheme we consider the following example. Suppose in a pandemic, a vaccine developing company stores some characteristics in an encrypted form of the patients who are undergoing trials of a newly created vaccine. The authority wants to issue a decryption key that can be used by selected members of the company (e.g. a specific group of scientists and the members in the board of directors). The key only decrypts a specific statistical computation on the characteristics of patients that may help to determine the usability of the vaccine in a larger scale. However, such statistics should not be revealed to all the members and the secret-key should not be able to decrypt the whole data-set due to the welfare of the company. Therefore, we need to embed a policy (indicating the members who are eligible to learn) and a specific vector (which will be operated on the data-set to compute a specific statistic) into a single key that can be given to the selected members. In other words, we need to have attribute-based access control in IPFE scheme.

A natural solution to the above problem is given by the notion of functional encryption (FE) [15,32] which allows us to compute a secret-key  $\text{sk}_F$  corresponding to a function  $F$  that consists of a policy  $f^1$  and a vector  $\mathbf{y}$ . Given an encryption of message  $m = (att, \mathbf{x})$ , one learns  $F(m) = \langle \mathbf{x}, \mathbf{y} \rangle$  if  $f(att) = 0$ , using the secret-key  $\text{sk}_F$ . The indistinguishability security requires that an adversary should be unable to distinguish between encryptions of  $m_0 = (att, \mathbf{x}_0)$  and  $m_1 = (att, \mathbf{x}_1)$  even if it possesses many secret-keys for the functions  $F_1, \dots, F_n$  satisfying  $F_i(m_0) = F_i(m_1)$ , for all  $i$ . However, candidate FEs supporting the required function class exist from indistinguishable obfuscation (IO) or multilinear maps (Mmaps) [22,11] the security of which is not well-understood. While some candidate Mmaps (with degree  $\geq 2$ ) based constructions [23,10] are still conjectured to be secure, the other FE constructions relying on IO are currently going through a break-and-repair cycle [30,12]. Therefore, the security of existing FEs for general functions cannot be guaranteed from well-known standard assumptions. Looking into the current state of the art, it is more preferable to

---

<sup>1</sup> A policy is a boolean function and we say an input  $a$  satisfies the policy  $f$  if  $f(a) = 0$ .

construct FE for the needed functionality instead of focusing on FE for general function class. Our goal is to build an efficient FE scheme from standard assumption for a class containing only functions like  $F$  as described above. This motivation leads us to the following question.

*Is it possible to construct a public-key FE scheme where we can embed any boolean function along with a predicate vector into the secret-keys and encrypt a message vector with respect to an attribute so that decryption outputs the inner product between the predicate and message vectors only when the attribute satisfies the boolean function?*

**Our contribution.** To address the above concern, we present a primitive called *attribute-based IPFE* (ABIPFE) where policies are associated with the secret-keys and attributes are taken while encrypting messages. Our main contribution is a construction of such ABIPFE from Learning With Errors (LWE) assumption in the standard model. The policies can be represented by any polynomial-size bounded-depth boolean circuits and the size of secret-keys or ciphertext relies on the depth of the circuits. Our work takes inspiration from the framework of Abdalla et al. [3]. To obtain an ABIPFE supporting general class of policies, we devise a technique to combine the LWE-based ABE scheme of Boneh et al. [14] (which we call BGG<sup>+</sup>-ABE) and the LWE-based IPFE scheme of Agrawal et al. [6] (which is abbreviated as ALS-IPFE).

In an ABIPFE scheme, using a master secret-key  $\text{msk}$ , a central authority generates secret-keys of the form  $\text{sk}_{f,\mathbf{y}}$  for a tuple  $(f, \mathbf{y})$  where  $f$  is a depth- $d$  circuit and  $\mathbf{y}$  is a predicate vector that belongs to  $\mathbb{Z}_q^\ell$  for an integer (possibly prime)  $q$ . The sender uses the master public-key  $\text{mpk}$  to encrypt a message vector  $\mathbf{x} \in \mathbb{Z}_q^\ell$  with respect to an attribute  $\mathbf{a}$  which is a binary string of length  $k$  and produces a ciphertext  $\text{ct}$ . A receiver having  $\text{sk}_{f,\mathbf{y}}$ , can recover  $\langle \mathbf{x}, \mathbf{y} \rangle$  from  $\text{ct}$  if  $f(\mathbf{a}) = 0$ . We prove the co-selective indistinguishability (coSel-IND) of the ABIPFE where the adversary  $\mathcal{A}$  submits a challenge attribute  $\mathbf{a}^*$  and a function  $f^*$  such that  $f^*(\mathbf{a}^*) = 0$  before seeing  $\text{mpk}$ . However,  $\mathcal{A}$  can adaptively choose a polynomial number of predicate vectors  $\mathbf{y}$  and gets secret-keys of the form  $\text{sk}_{f^*,\mathbf{y}}$ . So,  $\mathcal{A}$  is given access to many secret-keys that can decrypt the challenge ciphertext. The adversary can also query a secret-key for  $(f, \mathbf{y})$  such that  $f(\mathbf{a}^*) = 1$ . If  $\mathbf{x}_0, \mathbf{x}_1$  are the challenge messages (which can be picked adaptively), we require that  $\langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle$  for all  $\mathbf{y}$  for which a secret-key  $\text{sk}_{f^*,\mathbf{y}}$  is released during key query phase. Note that using a standard complexity leveraging argument as in [13], we can also allow  $\mathcal{A}$  to choose the challenge attribute adaptively.

**Theorem 1** (Informal) *Assuming subexponential LWE, there exists a coSel-IND secure ABIPFE scheme with short secret-keys, the size of which depends on the maximum depth of the functions supported by the scheme.*

We show that our single input ABIPFE can be extended to a multi-input variant of ABIPFE which we call *attribute-based multi-input IPFE* (ABMIPFE) scheme. Suppose there are  $n$  encryption slots and each slot is associated with a single attribute  $\mathbf{a}_i$  which is linked to a party that belongs to the system. The  $i$ -th party can encrypt a vector  $\mathbf{x}_i$  with respect to  $\mathbf{a}_i$  to produce a ciphertext  $\text{ct}_i$ . The secret-

keys are associated to tuples of form  $(\{f_i, \mathbf{y}_i\}_{i=1}^n)$  which can be used to learn  $\sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{y}_i \rangle$  if  $f_i(\mathbf{a}_i) = 0$  for all  $i = 1, \dots, n$ . For security, we define a co-adaptive indistinguishability (coAdp-IND) notion where the adversary is forced to submit  $n$  functions  $f_1, \dots, f_n$  before setup whereas it can choose the predicate vectors  $(\{\mathbf{y}_i\}_{i=1}^n)$  adaptively for key queries. If  $\{\mathbf{x}_i^0, \mathbf{x}_i^1\}_{i=1}^n$  are the challenge messages then all the secret-key queries should satisfy  $\sum_{i=1}^n \langle \mathbf{x}_i^0, \mathbf{y}_i \rangle = \sum_{i=1}^n \langle \mathbf{x}_i^1, \mathbf{y}_i \rangle$ .

**Theorem 2** (Informal) *Assuming subexponential LWE, there exists a coAdp-IND secure ABMIPFE scheme with short secret-keys, the size of which depends on the maximum depth of the functions supported by the scheme and linear to the number of parties in the scheme.*

**Comparison to existing approaches.** We briefly compare our resulting IPFE schemes in reference to existing approaches. The notion of attribute-based functional encryption (ABFE) was formalized by Chen, Zang and Yiu [18] where they proposed a ciphertext-policy ABIPFE (CP-ABIPFE) scheme for limited functionality based on three decisional assumptions in bilinear groups of composite order. They prove the adaptive security in a comparatively weaker setting where the adversary is not allowed to query any secret-key that can decrypt the challenge ciphertext. Improving the security and efficiency, Abdalla et al. [3] gave constructions of CP-ABIPFE based on Decisional Diffie-Hellman (DDH) assumption in bilinear groups of prime order. They utilized the DDH-based IPFE of [6] and any ABE schemes that support dual-system encryption methodology [36] to achieve access control in IPFE setting that can mainly handle policies of equality testing, orthogonality testing, read-once monotone span programs whereas one of the appealing feature of our construction compared to these schemes is that we can embed any general policy represented by a boolean function into the secret-keys of our ABIPFE. The first construction of [3] is selectively secure in simulation setting and the other is adaptively secure in indistinguishability setting. Both of these schemes allow the adversary to have many secret-keys for different attributes that can decrypt the challenge ciphertext, but the advantage of the adversary grows linearly with the number of secret-key queries. In the same work, they also present a natural extension of their pairing-based CP-ABIPFEs to MIPFEs using a generic transformation originally presented in the work of [2]. In this context, it is worth mentioning that our ABIPFE and ABMIPFE are based on standard LWE assumption and hence they are post-quantum secure.

The second main contribution of [3] is the constructions of two adaptively secure identity-based IPFE (IBIPFE) schemes based on the hardness of LWE problem. They combined the ALS-IPFE with two existing LWE-based IBEs. The first one uses the IBE from [25] to get a scheme secure in the random oracle model and the second one relies on the IBE from [5] to obtain a scheme secure in the standard model. In another work [21], Dufour-Sans and Pointcheval built a selectively secure identity-based FE scheme for unbounded inner product functionality in the random-oracle model under Bilinear Decisional Diffie-Hellman assumption. The main advantage of their scheme is the constant size master public-key and secret-keys, in particular, each of them consists of only one group

element. Compared to all these IPFE schemes, our IPFE undoubtedly provides a much finer access control that covers almost all practical applications.

In the context of constructing indistinguishability obfuscation, the authors of [9,27,28] built a primitive called *restricted* FE (latterly renamed as *partially-hiding* FE or PHFE) where the supported function class can execute degree-2 computation on its private input and offers a variety of computations on the public input such as degree-2 functions,  $\text{NC}_0$  or  $\text{NC}_1$  circuits. While all these PHFEs are described in secret-key setting, recently in [24], the authors proposed a public-key PHFE scheme supporting degree-2 functions in the private input and arithmetic  $\text{NC}_1$  functions over the public attribute. The PHFEs are proven secure relying on pairing-based assumptions. On the other hand, our ABIPFE is the first to support any polynomial-size boolean functions over the attributes in public-key setting with security based on standard LWE assumption.

**Technical overview.** The starting point is the IBIPFE construction of Abdalla et al. [3] where secret-keys and ciphertexts need to be associated with the same identity for a successful decryption. We use  $\text{BGG}^+$ -ABE and ALS-IPFE to build our ABIPFE and its multi-input variant. The challenge comes in controlling the noise in the ciphertexts for correct decryption and handling secret-key queries that decrypts the challenge ciphertext. We briefly describe the technical road towards achieving this goal. Our core approach utilizes the homomorphic evaluation procedure of [14] which can handle any polynomial-size bounded-depth (unbounded fan-in) boolean circuits of the form  $f : \{0,1\}^k \rightarrow \{0,1\}$ . Given matrices  $\vec{\mathbf{B}} = (\mathbf{B}_1, \dots, \mathbf{B}_k)$ , there are encoding mechanisms such that for any  $\mathbf{a} \in \{0,1\}^k$  and function  $f$  we have  $\mathbf{B}_a \leftarrow \text{Encode}_a(\vec{\mathbf{B}}, \mathbf{a})$  and  $\mathbf{B}_f \leftarrow \text{Encode}_f(\vec{\mathbf{B}}, f)$ . When a dual Regev encryption (as described in [33,25])  $\mathbf{c}_a = \mathbf{B}_a^\top \mathbf{s} + \text{noise}$  with respect to the public matrix  $\mathbf{B}_a$  is available, one can apply a conversion algorithm  $\text{Convert}_{\text{ct}}$  to compute  $\text{Convert}_{\text{ct}}(\mathbf{c}_a, \mathbf{a}, f) = \mathbf{B}_f^\top \mathbf{s} + \text{noise}'$  whenever  $f(\mathbf{a}) = 0$ . The master public-key  $\text{mpk}$  of our ABIPFE consists of matrices  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ ,  $\vec{\mathbf{B}} \in (\mathbb{Z}_q^{n \times m})^k$ ,  $\mathbf{D} \in \mathbb{Z}_q^{n \times \ell}$  and the master secret-key is a short basis  $\mathbf{T}_A$  of the lattice  $\Lambda_q^\perp(\mathbf{A})$ . To generate a secret-key  $\text{sk}_{f,\mathbf{y}}$  for a tuple  $(f, \mathbf{y})$ , the authority first computes  $\mathbf{B}_f \leftarrow \text{Encode}_f(\vec{\mathbf{B}}, f)$  and generates a low-norm matrix  $\mathbf{R}_f$  using  $\mathbf{T}_A$  such that  $(\mathbf{A}|\mathbf{B}_f) \cdot \mathbf{R}_f = \mathbf{D}$ . Finally, it sets  $\text{sk}_{f,\mathbf{y}} = \mathbf{R}_f \cdot \mathbf{y}$ .

An encryption of a message vector  $\mathbf{x} \in \mathbb{Z}_q^\ell$  with respect to an attribute  $\mathbf{a} \in \{0,1\}^k$  proceeds to compute  $\mathbf{B}_a \leftarrow \text{Encode}_a(\vec{\mathbf{B}}, \mathbf{a})$  and a Dual-Regev encryption  $(\mathbf{c}_0 = \mathbf{A}^\top \mathbf{s} + \mathbf{e}_0, \mathbf{c}_a = \mathbf{B}_a^\top \mathbf{s} + \mathbf{e}_1)$ . It encrypts the message as  $\mathbf{c} = \mathbf{D}^\top \mathbf{s} + \mathbf{e}_2 + \mathbf{x}$ . Here,  $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2$  denote the noise vectors. The ciphertext  $\text{ct}$  consists of  $(\mathbf{c}_0, \mathbf{c}_a, \mathbf{c})$ .

A receiver holding a secret-key  $\text{sk}_{f,\mathbf{y}}$  such that  $f(\mathbf{a}) = 0$  first obtains  $\mathbf{c}_f = \text{Convert}_{\text{ct}}(\mathbf{c}_a, \mathbf{a}, f)$  and then computes the inner product as

$$\begin{aligned} \mathbf{y}^\top \mathbf{c} - \text{sk}_{f,\mathbf{y}}^\top (\mathbf{c}_0 | \mathbf{c}_f) &\approx (\mathbf{D}\mathbf{y})^\top \mathbf{s} + \mathbf{y}^\top \mathbf{x} - (\mathbf{R}_f \cdot \mathbf{y})^\top (\mathbf{A}|\mathbf{B}_f)^\top \mathbf{s} \\ &= (\mathbf{D}\mathbf{y})^\top \mathbf{s} + \langle \mathbf{x}, \mathbf{y} \rangle - \mathbf{y}^\top ((\mathbf{A}|\mathbf{B}_f) \cdot \mathbf{R}_f)^\top \mathbf{s} = \langle \mathbf{x}, \mathbf{y} \rangle \end{aligned}$$

We prove  $\text{coSel-IND}$  security for our ABIPFE scheme using the proof techniques of  $\text{BGG}^+$ -ABE and IBIPFE scheme of [3]. The main technical difference is to

program the public matrix  $\mathbf{D}$  in such a way that we can generate secret-keys for a fixed function  $f$  while varying the associated predicate vectors without using  $\text{msk}$ . In other words, we need to generate a matrix  $\mathbf{Z}$  satisfying  $(\mathbf{A}|\mathbf{B}_f)\mathbf{Z} = \mathbf{D}$  such that each row of  $\mathbf{Z}$  follows the same distribution  $\mathcal{D}$  as that of  $\mathbf{R}_f$ . For that, we first pick a matrix  $\mathbf{Z}_1$  whose rows are coming from  $\mathcal{D}$  and define a matrix  $\mathbf{D}_1 = \mathbf{A}\mathbf{Z}_1$ . Then, we choose another matrix  $\mathbf{Z}_2$  following the same distribution as of  $\mathbf{Z}_1$  and set  $\mathbf{D} = \mathbf{D}_1 + \mathbf{B}_f\mathbf{Z}_2$ . Since  $\mathbf{Z}_1$  is a low-norm matrix,  $\mathbf{D}_1 = \mathbf{A}\mathbf{Z}_1$  is uniformly distributed over  $\mathbb{Z}_q^{n \times \ell}$  by a left-over hash lemma [5]. This ensures that  $\mathbf{D}$  is also uniform over  $\mathbb{Z}_q^{n \times \ell}$  and we can set  $\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix}$  which is distributed according to  $\mathbf{R}_f$ . We can now generate a secret-key  $\text{sk}_{f,\mathbf{y}}$  for any vector  $\mathbf{y}$  as  $\mathbf{Z} \cdot \mathbf{y}$ . The secret matrix  $\mathbf{Z}_1$  plays the role of master secret-key in the ALS-IPFE scheme when we finally depend on the hardness of LWE problem to conclude the security of our scheme.

We convert any single-input ABIPFE into an ABMIPFE via a generic transformation inspired from the works of Abdalla et al. [2,3] where they generically convert an IPFE into a multi-input IPFE (MIPFE) without using any additional primitive. The fact that our ABIPFE satisfies certain additional structural properties, namely *two-step decryption* and *linear encryption* [2], helps us to build the *first* ABMIPFE based on LWE assumption.

## 2 Preliminaries

**Notations.** For  $n \in \mathbb{N}$ , we denote by  $[n]$  the set  $\{1, \dots, n\}$ . We denote by  $x \leftarrow \mathcal{D}$  the process of sampling a value  $x$  according to the distribution of  $\mathcal{D}$ . We consider  $x \leftarrow S$  as the process of random sampling of a value  $x$  according to the uniform distribution over a finite set  $S$ . We denote by  $\mathbf{A} \otimes \mathbf{B}$  the tensor product between the matrices  $\mathbf{A}$  and  $\mathbf{B}$ . The inner product between two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^\ell$  is written as  $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{\ell} x_i y_i = \mathbf{y}^T \mathbf{x}$ . For any  $\lambda > \lambda_0$ , if a non-negative function  $\text{negl}$  satisfies  $\text{negl}(\lambda) < 1/\lambda^c$ ,  $c$  is a constant, then  $\text{negl}$  is called a *negligible* function over the positive integers.

### 2.1 Attribute-Based Inner Product Functional Encryption

An attribute-based inner product functional encryption (ABIPFE) scheme for a class of functions  $\mathcal{F}_\lambda = \{f : \mathcal{S}_\lambda \rightarrow \{0, 1\}\}$ , a predicate space  $\mathcal{Y}_\lambda$  and a message space  $\mathcal{X}_\lambda$  consists of four probabilistic polynomial time (PPT) algorithms  $\text{ABIPFE} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$  satisfying the following requirement:

- $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda)$ : The setup algorithm on input a security parameter  $\lambda$ , a vector length parameter  $\ell$  and a function class  $\mathcal{F}_\lambda$ , outputs a master public-key  $\text{mpk}$  and a master secret-key  $\text{msk}$ .
- $\text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{mpk}, \text{msk}, f, \mathbf{y})$ : The key generation algorithm takes as input the key pairs  $(\text{mpk}, \text{msk})$ , a function  $f \in \mathcal{F}_\lambda$  and a vector  $\mathbf{y} \in \mathcal{Y}_\lambda$  of length  $\ell$ . It outputs a secret-key  $\text{sk}_{f,\mathbf{y}}$  which also includes the description of  $f$  and the vector  $\mathbf{y}$ .

- $\text{ct} \leftarrow \text{Enc}(\text{mpk}, a, \mathbf{x})$ : The encryption algorithm takes input the master public-key  $\text{mpk}$ , an attribute  $a \in \mathcal{S}_\lambda$  and a message vector  $\mathbf{x} \in \mathcal{X}_\lambda$ . It outputs a ciphertext  $\text{ct}$  which contains the attribute  $a$ .
- $\perp$  or  $\zeta \leftarrow \text{Dec}(\text{mpk}, \text{sk}_{f,\mathbf{y}}, \text{ct})$ : The decryption algorithm is deterministic. It takes as input the master public-key  $\text{mpk}$ , a secret-key  $\text{sk}_{f,\mathbf{y}}$  and a ciphertext  $\text{ct}$ . It outputs either a message  $\zeta \in \mathbb{Z}$  or a symbol  $\perp$  indicating failure.

**Definition 1 (Correctness)** An ABIPFE is said to be correct if for all  $\lambda \in \mathbb{N}$ ,  $f \in \mathcal{F}_\lambda$ ,  $\mathbf{y} \in \mathcal{Y}_\lambda$ ,  $a \in \mathcal{S}_\lambda$ ,  $\mathbf{x} \in \mathcal{X}_\lambda$  we have

$$\Pr \left[ \begin{array}{l} \langle \mathbf{x}, \mathbf{y} \rangle = \text{Dec}(\text{mpk}, \text{sk}_{f,\mathbf{y}}, \text{ct}) \\ \wedge \quad f(a) = 0 \end{array} : \begin{array}{l} (\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda), \\ \text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{mpk}, \text{msk}, f, \mathbf{y}), \\ \text{ct} \leftarrow \text{Enc}(\text{mpk}, a, \mathbf{x}) \end{array} \right] = 1 - \text{negl}(\lambda)$$

where the probability is taken over the random coins of  $\text{Setup}$ ,  $\text{KeyGen}$  and  $\text{Enc}$ .

We define  $Q$ -bounded  $\text{coSel-IND}$  security for ABIPFE. Let  $a^* \in \mathcal{S}_\lambda$  be the target attribute. We call  $f$  a *target accepting* function if  $f(a^*) = 0$ . In  $Q$ -bounded  $\text{coSel-IND}$  game, the adversary  $\mathcal{A}$  submits the target attribute  $a^*$  and  $Q$  target accepting functions before seeing  $\text{mpk}$ . Note that,  $\mathcal{A}$  is allowed to adaptively choose associated predicate vectors and functions which output 1 on input  $a^*$ .

**Definition 2 ( $Q$ -bounded  $\text{coSel-IND}$  security for ABIPFE)** For an ABIPFE scheme  $\text{ABIPFE} = (\text{Setup}, \text{Keygen}, \text{Enc}, \text{Dec})$  for a function family  $\mathcal{F}_\lambda$ , a predicate space  $\mathcal{Y}_\lambda$ , an attribute space  $\mathcal{S}_\lambda$ , a message space  $\mathcal{X}_\lambda$  and for any PPT adversary  $\mathcal{A}$ , we define  $Q$ -bounded  $\text{coSel-IND}$  security experiment  $\text{Expt}_{\mathcal{A}, \text{ABIPFE}}^{\text{coSel-IND}}(1^\lambda)$  as follows.

1. **Pre-Setup Phase.** The adversary  $\mathcal{A}$  on input  $1^\lambda$ , outputs a target attribute  $a^* \in \mathcal{S}_\lambda$  and a set  $\{f_1, \dots, f_Q\}$  of  $Q$  target accepting functions.
2. **Setup Phase.** On input  $1^\lambda, 1^\ell$  and  $\mathcal{F}_\lambda$ , the challenger samples  $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda)$ . It gives  $\text{mpk}$  to  $\mathcal{A}$ .
3. **Query Phase.** During the experiment  $\mathcal{A}$  can make the following queries in any arbitrary order.  $\mathcal{A}$  can make unbounded many key queries, however, it is allowed to make only one challenge query.
  - (a) **Key Queries.**  $\mathcal{A}$  sends  $(f, \mathbf{y}) \in \mathcal{F}_\lambda \times \mathcal{Y}_\lambda$  and the challenger returns  $\text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{mpk}, \text{msk}, f, \mathbf{y})$ .
  - (b) **Challenge Query.**  $\mathcal{A}$  submits a pair of messages  $(\mathbf{x}_0, \mathbf{x}_1) \in \mathcal{X}_\lambda^2$ . The challenger samples a bit  $b \leftarrow \{0, 1\}$  and returns  $\text{ct} \leftarrow \text{Enc}(\text{mpk}, a^*, \mathbf{x}_b)$ . We require that any secret-key query  $(f_j, \mathbf{y}_j)$  should satisfy  $(j \in [Q] \wedge \langle \mathbf{x}_0, \mathbf{y}_j \rangle = \langle \mathbf{x}_1, \mathbf{y}_j \rangle)$  or  $f_j(a^*) = 1$ .
4. **Guess Phase.**  $\mathcal{A}$  outputs a guess bit  $b'$ . The experiment outputs 1 if  $b = b'$ .

The ABIPFE is said to satisfy  $Q$ -bounded  $\text{coSel-IND}$  security (or simply co-selective security when  $Q$  is clear from the context) if the advantage

$$\text{Adv}_{\mathcal{A}, \text{ABIPFE}}^{\text{coSel-IND}}(\lambda) = \left| \Pr[\text{Expt}_{\mathcal{A}, \text{ABIPFE}}^{\text{coSel-IND}}(1^\lambda) = 1] - \frac{1}{2} \right|$$

of  $\mathcal{A}$  in the above game is negligible in  $\lambda$ .

We can also define stronger versions of the security such as selective and adaptive experiments. In Sel-IND security game the adversary  $\mathcal{A}$  submits the target attribute  $a^*$  in the pre-setup phase and it is allowed to choose the target accepting functions adaptively. We give more power to  $\mathcal{A}$  in the Adp-IND security experiment. In particular,  $\mathcal{A}$  has the freedom to choose the target attribute  $a^*$  in the challenge phase and target accepting functions in the key query phase. Accordingly, we can define the advantages by the functions  $\text{Adv}_{\mathcal{A}, \text{ABIPFE}}^{\text{Sel-IND}}(\lambda)$  and  $\text{Adv}_{\mathcal{A}, \text{ABIPFE}}^{\text{Adp-IND}}(\lambda)$  in the selective and adaptive security experiments respectively.

## 2.2 Lattice Preliminaries [14,3]

We recall basics of lattices and some important results related to our construction of ABIPFE. Let  $n, m, q$  be positive integers such that  $n = \text{poly}(\lambda)$  and  $m \geq n \lceil \log q \rceil$ . For a matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ , we let  $\Lambda_q^\perp(\mathbf{A})$  denotes the lattice  $\{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \text{ in } \mathbb{Z}_q\}$ . More generally for  $\mathbf{u} \in \mathbb{Z}_q^n$ , we let  $\Lambda_q^{\mathbf{u}}(\mathbf{A})$  denote the lattice  $\{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{u} \text{ in } \mathbb{Z}_q\}$ . For a lattice  $\Lambda$  of dimension  $n$ , we denote  $\Lambda^* = \{\mathbf{u} \in \mathbb{R}^n : \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z} \text{ for all } \mathbf{v} \in \Lambda\}$  by the dual lattice.

**Matrix norms.** For a vector  $\mathbf{u}$ , we let  $\|\mathbf{u}\|$  denote its  $\ell_2$  norm. For a matrix  $\mathbf{R} \in \mathbb{Z}^{k \times m}$ , let  $\tilde{\mathbf{R}}$  be the result of applying Gram-Schmidt (GS) orthogonalization to the columns of  $\mathbf{R}$ . We define the following norms.

- $\|\mathbf{R}\|$  denotes the  $\ell_2$  norm of the longest column of  $\mathbf{R}$ .
- $\|\mathbf{R}\|_2$  denotes the operator norm of  $\mathbf{R}$  defined as  $\|\mathbf{R}\|_2 = \sup_{\|\mathbf{x}\|=1} \|\mathbf{R}\mathbf{x}\|$ .
- $s_1(\mathbf{R})$  denotes the spectral norm of  $\mathbf{R}$  (largest singular value of  $\mathbf{R}$ ).

In addition, we know that  $\|\tilde{\mathbf{R}}\| \leq \|\mathbf{R}\| \leq \|\mathbf{R}\|_2 \leq \sqrt{k}\mathbf{R}$ . The spectral norm of concatenating matrices are bounded as  $s_1(\mathbf{R}|\mathbf{S}) \leq \sqrt{s_1(\mathbf{R})^2 + s_1(\mathbf{S})^2}$ . The following lemma provides a bound on spectral norm.

**Lemma 1** [20] *Let  $\mathbf{X} \in \mathbb{R}^{n \times m}$  be a sub-Gaussian random matrix with parameter  $s$ . There exists a universal constant  $C \approx \frac{1}{\sqrt{2\pi}}$  such that for any  $t \geq 0$ , we have  $s_1(\mathbf{X}) \leq C \cdot s \cdot (\sqrt{m} + \sqrt{n} + t)$  except with probability at most  $2 \cdot \exp(-\pi t^2)$ .*

**Lemma 2 (Gram-Schmidt minimum [16])** *For any arbitrary  $n$ -dimensional integer lattice  $\Lambda$ , it holds that:*

$$1 \leq \lambda_1(\Lambda^*) \cdot \min_{\mathbf{B}} \|\tilde{\mathbf{B}}\| \leq \gamma^2 n,$$

where the minimum is over all (ordered) bases  $\mathbf{B}$  of lattice  $\Lambda$  and  $\gamma$  is a constant.

**Gaussian distribution.** For any  $n$ -dimensional lattice  $\Lambda$ , the discrete Gaussian distribution over  $\Lambda$  with center  $\mathbf{c} \in \mathbb{R}^n$  and parameter  $\sigma > 0$  is defined as  $\mathcal{D}_{\Lambda, \sigma, \mathbf{c}}(\mathbf{x}) = \rho_{\sigma, \mathbf{c}}(\mathbf{x}) / \rho_{\sigma, \mathbf{c}}(\Lambda)$ ,  $\forall \mathbf{x} \in \mathbb{R}^n$  where  $\rho_{\sigma, \mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|_2^2 / \sigma^2)$  and  $\rho_{\sigma, \mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma, \mathbf{c}}(\mathbf{x})$ . When  $\mathbf{c} = 0$ , we use  $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{u}}(\mathbf{A}))$  for a parameter  $\sigma > 0$  to denote a discrete Gaussian distribution over the lattice  $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ . For a random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  and  $\sigma = \tilde{\Omega}(\sqrt{n})$ , a vector  $\mathbf{x}$  sampled from  $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{u}}(\mathbf{A}))$  has  $\ell_2$  norm less than  $\sigma\sqrt{m}$  with probability at least  $1 - \text{negl}(m)$ . For a matrix  $\mathbf{U} = (\mathbf{u}_1 | \dots | \mathbf{u}_k) \in \mathbb{Z}_q^{n \times k}$ , we let  $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$  be a distribution on matrices in



$\mathbb{Z}^{m \times k}$  where the  $i$ -th column is sampled from  $\mathcal{D}_\sigma(\Lambda_q^{u_i}(\mathbf{A}))$  independently for  $i = 1, \dots, k$ . Clearly if  $\mathbf{R}$ , is sampled from  $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$  then  $\mathbf{AR} = \mathbf{U}$  in  $\mathbb{Z}_q$ .

**Learning with errors(LWE) [33].** Fix integers  $n, m$ , a prime integer  $q$  and a noise distribution  $\chi$  over  $\mathbb{Z}$ . The  $\text{LWE}_{q,\chi,n}$  problem is to distinguish between the distributions  $(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + e)$  and  $(\mathbf{A}, \mathbf{u})$  where  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ ,  $\mathbf{s} \in \mathbb{Z}_q^n$ ,  $\mathbf{u} \in \mathbb{Z}_q^m$  are independently sampled.

**Proposition 1 [33]** *Let  $\alpha = \alpha(n) \in (0, 1)$  and let  $q = q(n)$  be a prime such that  $\alpha \cdot q > 2\sqrt{n}$ . If there exists an efficient (possibly quantum) algorithm that solves  $\text{LWE}_{q,\Psi_\alpha}$ , then there exists an efficient quantum algorithm for approximating SIVP and GapSVP in the  $\ell_2$  norm, in the worst case, to within  $\tilde{O}(n/\alpha)$  factors.*

Here  $\Psi_\alpha$  is distributed as  $\lceil qX \rceil \bmod q$  where  $X$  is a normal random variable with mean 0 and standard deviation  $\alpha/\sqrt{2\pi}$ .

**Solving  $\mathbf{AZ} = \mathbf{U}$ .** We review algorithms for finding a low-norm matrix  $\mathbf{Z} \in \mathbb{Z}_q^{m \times k}$  such that  $\mathbf{AZ} = \mathbf{U}$ .

**Theorem 3 [25]** *There is a PPT SampleD that, given a basis  $\mathbf{B}$  of an  $n$ -dimensional lattice  $\Lambda = \mathcal{L}(\mathbf{B})$ , a parameter  $\sigma \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$  and a center  $\mathbf{c} \in \mathbb{R}^n$ , outputs a sample from a distribution that is statistically close to  $\mathcal{D}_{\sigma,\mathbf{c}}(\Lambda)$ .*

**Proposition 2 [7]** *For any prime  $q = \text{poly}(n)$  and any  $m \geq 5n \lg q$ , there is a probabilistic polynomial-time algorithm SampleMat that, on input  $1^n$ , outputs a matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  and a full-rank set  $\mathbf{S} \subset \Lambda_q^\perp(\mathbf{A})$ , where the distribution of  $\mathbf{A}$  is statistically close to uniform over  $\mathbb{Z}_q^{n \times m}$  and the length  $\|\mathbf{S}\| \leq L = m^{2.5}$ .*

Also,  $\mathbf{S}$  can be converted efficiently to a “good” basis  $\mathbf{T}$  of  $\Lambda_q^\perp(\mathbf{A})$  such that  $\|\tilde{\mathbf{T}}\| \leq \|\tilde{\mathbf{S}}\| \leq L$ .

**Lemma 3 (Preimage samplable functions [25])** *For any prime  $q = \text{poly}(n)$ , any  $m \geq 6n \log q$ , and any  $\sigma \geq L \cdot \omega(\sqrt{\log m})$ , it holds that there exists PPT algorithms TrapGen, SampleD, SamplePre such that:*

1. TrapGen computes  $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$ , where  $\mathbf{A}$  is statistically close to uniform over  $\mathbb{Z}_q^{n \times m}$  and  $\mathbf{T}_\mathbf{A} \subset \Lambda_q^\perp(\mathbf{A})$  is a good basis with  $\|\tilde{\mathbf{T}}_\mathbf{A}\| \leq L$ . The matrix  $\mathbf{A}$  is public and  $\mathbf{T}_\mathbf{A}$  is the trapdoor.
2. SampleD is used to sample vectors from  $\mathcal{D}_\sigma(\mathbb{Z}^{m \times k})$ .
3. The trapdoor inversion algorithm SamplePre( $\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{U}, \sigma$ ) outputs a matrix  $\mathbf{Z} \in \mathbb{Z}^{m \times k}$  such that  $\mathbf{AZ} = \mathbf{U}$ .

In addition, it holds that the following distributions are statistically close:

$$\text{Dist}_1 := (\mathbf{A}, \mathbf{Z}, \mathbf{U}) \text{ s.t. } (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q), \mathbf{U} \leftarrow \mathbb{Z}^{n \times k}, \\ \mathbf{Z} \leftarrow \text{SamplePre}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{U}, \sigma)$$

$$\text{Dist}_2 := (\mathbf{A}, \mathbf{Z}, \mathbf{AZ}) \text{ s.t. } \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{Z} \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^{m \times k}) : \|\mathbf{z}_i\| \leq \sigma\sqrt{m}, i \in [k], \\ \text{where } \mathbf{z}_i \text{ is the } i\text{-th column of } \mathbf{Z}$$

**Trapdoor generators.** The following Lemma states properties of algorithms for generating short basis of lattices.

**Lemma 4** [14] *Let  $n, m, q > 0$  be integers with  $q$  prime. There are polynomial-time algorithms with the properties below:*

1.  $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$  ([7,8,31]): a randomized algorithm that, when  $m = \Theta(n \log q)$ , outputs a full-rank matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  and basis  $\mathbf{T}_\mathbf{A} \in \mathbb{Z}^{m \times m}$  for  $\Lambda_q^\perp(\mathbf{A})$  such that  $\mathbf{A}$  is  $\text{negl}(n)$ -close to uniform and  $\|\tilde{\mathbf{T}}_\mathbf{A}\| = O(\sqrt{n \log q})$ , with all but negligible probability in  $n$ .
2.  $\mathbf{T}_{\mathbf{A}|\mathbf{B}} \leftarrow \text{ExtendRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B})$  ([17]): a deterministic algorithm that given full-rank matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$  and a basis  $\mathbf{T}_\mathbf{A}$  of  $\Lambda_q^\perp(\mathbf{A})$  outputs a basis  $\mathbf{T}_{\mathbf{A}|\mathbf{B}}$  of  $\Lambda_q^\perp(\mathbf{A}|\mathbf{B})$  such that  $\|\tilde{\mathbf{T}}_\mathbf{A}\| = \|\tilde{\mathbf{T}}_{\mathbf{A}|\mathbf{B}}\|$ .
3.  $\mathbf{T}_\mathbf{H} \leftarrow \text{ExtendLeft}(\mathbf{A}, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{S})$  where  $\mathbf{H} = (\mathbf{A}|\mathbf{G} + \mathbf{A}\mathbf{S})$  ([5]): a deterministic algorithm that given full-rank matrices  $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n \times m}$  and a basis  $\mathbf{T}_\mathbf{G}$  of  $\Lambda_q^\perp(\mathbf{G})$  outputs a basis  $\mathbf{T}_\mathbf{H}$  of  $\Lambda_q^\perp(\mathbf{H})$  such that  $\|\tilde{\mathbf{T}}_\mathbf{H}\| = \|\tilde{\mathbf{T}}_\mathbf{G}\| \cdot (1 + \|\mathbf{S}\|_2)$
4. For  $m = n \lceil \log q \rceil$  there is a fixed full-rank matrix  $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$  such that the lattice  $\Lambda_q^\perp(\mathbf{G})$  has a publicly known basis  $\mathbf{T}_\mathbf{G} \in \mathbb{Z}^{m \times m}$  with  $\|\tilde{\mathbf{T}}_\mathbf{G}\| \leq \sqrt{5}$ .

**Lemma 5** [5,17] *Let  $n, m, \ell, q > 0$  be integers with  $q$  prime. There exist the following polynomial-time algorithms.*

1.  $\mathbf{Z} \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}, \mathbf{U}, \sigma)$ : a randomized algorithm that given full-rank matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$ , matrix  $\mathbf{U} \in \mathbb{Z}_q^{n \times \ell}$ , a basis  $\mathbf{T}_\mathbf{A}$  of  $\Lambda_q^\perp(\mathbf{A})$  and  $\sigma \geq \|\tilde{\mathbf{T}}_\mathbf{A}\| \cdot \omega(\sqrt{\log m})$ , outputs a random sample  $\mathbf{Z} \in \mathbb{Z}_q^{2m \times \ell}$  from a distribution that is statistically close to  $\mathcal{D}_\sigma(\Lambda_q^\mathbf{U}(\mathbf{A}|\mathbf{B}))$ . This algorithm is the composition of two algorithms:  $\mathbf{T}_{\mathbf{A}|\mathbf{B}} \leftarrow \text{ExtendRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B})$  and  $\mathbf{Z} \leftarrow \text{SamplePre}((\mathbf{A}|\mathbf{B}), \mathbf{T}_{\mathbf{A}|\mathbf{B}}, \mathbf{U}, \sigma)$ .
2.  $\mathbf{Z} \leftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{S}, y, \mathbf{U}, \sigma)$ : a randomized algorithm that given full-rank matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ , matrices  $\mathbf{S} \in \mathbb{Z}_q^{m \times m}$ ,  $\mathbf{U} \in \mathbb{Z}_q^{n \times \ell}$ ,  $y \neq 0 \in \mathbb{Z}_q$  and  $\sigma \geq \sqrt{5} \cdot (1 + \|\mathbf{S}\|_2) \cdot \omega(\sqrt{\log m})$ , outputs a random sample  $\mathbf{Z} \in \mathbb{Z}_q^{2m \times \ell}$  from a distribution that is statistically close to  $\mathcal{D}_\sigma(\Lambda_q^\mathbf{U}(\mathbf{A}|y\mathbf{G} + \mathbf{A}\mathbf{S}))$ . This algorithm is the composition of two algorithms:  $\mathbf{T}_{(\mathbf{A}|y\mathbf{G} + \mathbf{A}\mathbf{S})} \leftarrow \text{ExtendLeft}(\mathbf{A}, y\mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{S})$  and  $\mathbf{Z} \leftarrow \text{SamplePre}((\mathbf{A}|y\mathbf{G} + \mathbf{A}\mathbf{S}), \mathbf{T}_{(\mathbf{A}|y\mathbf{G} + \mathbf{A}\mathbf{S})}, \mathbf{U}, \sigma)$ .

**Randomness extraction.** We consider a version of left-over hash lemma.

**Lemma 6** [5] *Suppose that  $m > (n + 1) \log_2 q + \omega(\log n)$  and that  $q > 2$  is a prime. Let  $\mathbf{S}$  be an  $m \times k$  matrix chosen uniformly in  $\{\pm 1\}^{m \times k} \bmod q$  where  $k = k(n)$  is a polynomial in  $n$ . Let  $\mathbf{A}$  and  $\mathbf{B}$  be matrices chosen uniformly in  $\mathbb{Z}_q^{n \times m}$  and  $\mathbb{Z}_q^{n \times k}$  respectively. Then, for all vectors  $\mathbf{e} \in \mathbb{Z}_q^m$ , the distribution  $(\mathbf{A}, \mathbf{A}\mathbf{S}, \mathbf{S}^\top \mathbf{e})$  is statistically close to the distribution  $(\mathbf{A}, \mathbf{B}, \mathbf{S}^\top \mathbf{e})$ .*

Note that the Lemma holds for every vector  $\mathbf{e}$  in  $\mathbb{Z}_q^m$  including low norm vectors.

**Noise rerandomization.** We describe the algorithm  $\text{NoiseGen}(\mathbf{R}, s)$  from [29]. On input a matrix  $\mathbf{R} \in \mathbb{Z}^{m \times t}$  and  $s \in \mathbb{R}^+$  such that  $s > s_1(\mathbf{R}\mathbf{R}^\top)$ , it first

samples  $\mathbf{e}_1 := \mathbf{R}\mathbf{e} + (s^2\mathbf{I}_m - \mathbf{R}\mathbf{R}^\top)^{\frac{1}{2}}\mathbf{e}'$ , where  $\mathbf{I}_m$  denotes the identity matrix of order  $m$ , and  $\mathbf{e} \leftarrow \mathcal{D}_\sigma^t, \mathbf{e}' \leftarrow \mathcal{D}_{\sqrt{2}\sigma}^m$  are independent spherical continuous Gaussian noises. Then, it samples  $\mathbf{e}_2 \leftarrow \mathcal{D}_{s\sqrt{2}\sigma}(\mathbb{Z}^m - \mathbf{e}_1)$ , and returns  $\mathbf{e}_1 + \mathbf{e}_2$ . We have the following Lemma:

**Lemma 7 (Noise distribution [29])** . *Let  $\mathbf{R} \leftarrow \mathbb{Z}^{m \times t}$  and  $s > s_1(\mathbf{R})$ . Then, for all vectors  $\mathbf{e} \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^t)$ , the distribution of  $\mathbf{R}\mathbf{e} + \text{NoiseGen}(\mathbf{R}, s)$  is statistically close to  $\mathcal{D}_{2s\sigma}(\mathbb{Z}^m)$ .*

### 2.3 Homomorphic Evaluation Procedures.

We follow the abstraction of evaluation procedure in the LWE-based ABE scheme of [14]. Let  $n, m, k, q = q(n)$  be positive integers such that  $m = \Theta(n \log q)$  and  $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$  be a fixed matrix obtained by padding  $\mathbf{I}_n \otimes (1, 2, 4, 8, \dots, 2^{\lceil \log q \rceil})$  with zero columns.

**Theorem 4** *There exist efficient deterministic algorithms  $\text{Eval}_{\text{pk}}, \text{Eval}_{\text{ct}}, \text{Eval}_{\text{sim}}$  such that for any sequence of matrices  $(\mathbf{B}_1, \dots, \mathbf{B}_k) \in (\mathbb{Z}_q^{n \times m})^k$ , for any family of boolean functions  $\mathcal{F} = \{f : \{0, 1\}^k \rightarrow \{0, 1\}\}$  with maximum depth  $d$  and for every  $\mathbf{a} = (a_1, \dots, a_k) \in \{0, 1\}^k$ , the following properties hold:*

1.  $\mathbf{B}_f \leftarrow \text{Eval}_{\text{pk}}(f, (\mathbf{B}_1, \dots, \mathbf{B}_k))$  : On input a function  $f \in \mathcal{F}$  and matrices  $\{\mathbf{B}_i\}_{i \in [k]}$ , it outputs a matrix  $\mathbf{B}_f \in \mathbb{Z}_q^{n \times m}$ .
2.  $\mathbf{c}_f \leftarrow \text{Eval}_{\text{ct}}(f, ((a_i, \mathbf{B}_i, \mathbf{c}_i)_{i=1}^k))$  : On input a function  $f \in \mathcal{F}$ ,  $a_i \in \{0, 1\}$ ,  $\mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$  and  $\mathbf{c}_i \in \mathbb{Z}_q^m$  for  $i \in [k]$ , it outputs a vector  $\mathbf{c}_f \in \mathbb{Z}_q^m$  such that

$$\text{if } \{\mathbf{c}_i = (a_i\mathbf{G} + \mathbf{B}_i)^\top \mathbf{s} + \mathbf{e}_i\}_{i \in [k]} \text{ then } \mathbf{c}_f = (f(\mathbf{a})\mathbf{G} + \mathbf{B}_f)^\top \mathbf{s} + \mathbf{e}_f$$

where  $\mathbf{a} = (\mathbf{a}_1, \dots, \mathbf{a}_k) \in \{0, 1\}^k$  and  $\mathbf{B}_f = \text{Eval}_{\text{pk}}(f, (\mathbf{B}_1, \dots, \mathbf{B}_k))$ . Furthermore, we require that  $\|\mathbf{e}_f\| < \gamma_{\mathcal{F}} \cdot \max_{i \in [k]} \|\mathbf{e}_i\|$ .

3.  $\mathbf{S}_f \leftarrow \text{Eval}_{\text{sim}}(f, ((a_i, \mathbf{S}_i)_{i=1}^k, \mathbf{A}))$  : On input a function  $f \in \mathcal{F}$ ,  $a_i \in \{0, 1\}$ ,  $\mathbf{S}_i \in \{\pm 1\}^{m \times m}$  for  $i \in [k]$  and  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ , it outputs a matrix  $\mathbf{S}_f \in \mathbb{Z}_q^{m \times m}$  that satisfies

$$\mathbf{A}\mathbf{S}_f + f(\mathbf{a})\mathbf{G} = \mathbf{B}_f \text{ where } \mathbf{B}_f = \text{Eval}_{\text{pk}}(f, (\mathbf{A}\mathbf{S}_1 + a_1\mathbf{G}, \dots, \mathbf{A}\mathbf{S}_k + a_k\mathbf{G})).$$

Furthermore, we require that  $\|\mathbf{S}_f\|_2 \leq \gamma_{\mathcal{F}}$ .

For any family  $\mathcal{F}$  of depth- $d$  boolean functions the noise  $\gamma_{\mathcal{F}}$  (in worst case) is upper bounded by  $O(\sqrt{mm^d})$ .

## 3 Inner Product Functional Encryption Scheme of [6]

Before going to describe our ABIPFE construction in the next section, here we recall the ALS-IPFE construction of Agrawal et al. [6]. We describe a modified version of ALS-IPFE that was used in [3] to achieve identity-based IPFE. However, the modified version was developed in [4,35] where they simplify security parameter via a noise rerandomization technique of [29]. In the original

IPFE of [6], the secret matrix  $\mathbf{Z}$  was chosen from a distribution having a complex parameter setting whereas the security proof of [4,35] allows us to choose  $\mathbf{Z}$  from a simple discrete Gaussian distribution. We describe the modified ALS-IPFE for a predicate space  $\mathcal{Y}_\lambda = \{0, 1, \dots, V(\lambda) - 1\}^\ell$  and a message space  $\mathcal{X}_\lambda = \{0, 1, \dots, M(\lambda) - 1\}^\ell$ . Let us assume that inner product of any vector  $\mathbf{y} \in \mathcal{Y}_\lambda$  and any vector  $\mathbf{x} \in \mathcal{X}_\lambda$  is bounded by  $K = \ell VM$ .

**Setup**( $1^\lambda, 1^\ell$ ): On input  $1^\lambda$  and  $1^\ell$ , the setup algorithm samples  $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$ ,  $\mathbf{Z} \leftarrow \mathcal{D}_\rho(\mathbb{Z}^{\ell \times m})$  and sets  $\mathbf{D} = \mathbf{Z}\mathbf{A}$ . It returns the master public-key as  $\text{mpk} = (\mathbf{A}, \mathbf{D})$  and the master secret-key as  $\text{msk} = \mathbf{Z}$ .

**KeyGen**( $\text{msk}, \mathbf{y}$ ): On input  $\text{msk}$  and a vector  $\mathbf{y} \in \mathcal{Y}_\lambda$ , the algorithm returns a secret-key  $\text{sk}_\mathbf{y} = \mathbf{y}^\top \mathbf{Z} \in \mathbb{Z}$ .

**Enc**( $\text{mpk}, \mathbf{x}$ ): To encrypt a message  $\mathbf{x} \in \mathcal{X}_\lambda$  using  $\text{mpk}$  the encryption algorithm first samples  $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ ,  $\mathbf{e}_1 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^m)$ ,  $\mathbf{e}_2 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^\ell)$ . Then, it computes  $\text{ct}_1 = \mathbf{A}\mathbf{s} + \mathbf{e}_1$  and  $\text{ct}_2 = \mathbf{D}\mathbf{s} + \mathbf{e}_2 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}$ . Finally, it returns the ciphertext as  $\text{ct} = (\text{ct}_1, \text{ct}_2) \in \mathbb{Z}_q^{m+\ell}$ .

**Dec**( $\text{sk}_\mathbf{y}, \text{ct}$ ): The decryption algorithm parse the ciphertext  $\text{ct} = (\text{ct}_1, \text{ct}_2)$  and compute  $\zeta' = \mathbf{y}^\top \text{ct}_2 - \text{sk}_\mathbf{y} \cdot \text{ct}_1 \bmod q$ . It outputs  $\zeta \in \{0, 1, \dots, K-1\}$  which minimizes  $|\lfloor \frac{q}{K} \rfloor \cdot \zeta - \zeta'|$ .

IPFE can be treated as a particular case of ABIPFE when we assume that a secret-key is generated for a tuple  $(f, \mathbf{y})$  such that  $f(a) = 0$  for all  $a \in \mathcal{S}_\lambda$ . The Adp-IND security of IPFE allows an adversary  $\mathcal{A}$  to learn unbounded many secret-keys  $\text{sk}_\mathbf{y}$  for adaptively chosen predicate vectors  $\mathbf{y}$ . The secret-keys should satisfy  $\langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle$  where  $\mathbf{x}_0, \mathbf{x}_1$  are the (adaptively chosen) challenge messages. For correctness and security, we collect the following Lemma and Theorem developed in a series of work [6,4,35,3].

**Lemma 8 (Correctness of ALS-IPFE)** [6,4,35,3] *For  $\sigma, \rho > \omega(\sqrt{\log n})$  and  $q > 2K\ell\sqrt{\ell}V\omega(\log^2 n)$ , the ALS-IPFE scheme is correct.*

The adaptive security of ALS-IPFE is based on  $\text{LWE}_{q,\alpha,n}$  where the standard deviation of the noise distribution is  $\alpha q$ . The ALS-IPFE parameter setting of [3] is given by

$$q > 2K\ell\sqrt{\ell}V\omega(\log^2 n), \quad m = 2n \log q$$

$$\sigma = 2C'\alpha q(\sqrt{m} + \sqrt{n} + \sqrt{\ell}), \quad \rho \geq \omega(\sqrt{\log n})$$

where  $C'$  is a constant. When  $\alpha q > 2\sqrt{n}$ , with such parameter setting the  $\text{LWE}_{q,\alpha,n}$  is reducible to SIVP or GapSVP problem (Proposition 1).

**Theorem 5 (Security of ALS-IPFE)** [6,4,35,3] *Let  $n$  be the security parameter,  $m > 2n \log q$  and  $q, \sigma, \rho, \alpha \leq \frac{\sigma}{2C'q(\sqrt{m} + \sqrt{n} + \sqrt{\ell})}$  are as described above. Then, the ALS-IPFE scheme is Adp-IND secure, assuming  $\text{LWE}_{q,\alpha,n}$  is hard.*

## 4 Our Construction of ABIPFE from LWE

In this section, we describe our construction of an ABIPFE scheme based on the hardness of LWE problem in standard model. In particular, we use the ABE scheme of [14] and the ALS-IPFE scheme [6] as described in Sec. 3. We present our ABIPFE for a class of functions  $\mathcal{F}_\lambda = \{f : \{0, 1\}^k \rightarrow \{0, 1\}\}$ , a predicate space  $\mathcal{Y}_\lambda = \{0, \dots, V(\lambda) - 1\}^\ell$  and a message space  $\mathcal{X}_\lambda = \{0, \dots, X(\lambda) - 1\}^\ell$ . In addition, we assume that  $|\langle \mathbf{x}, \mathbf{y} \rangle| < K$  where  $K = \ell V X$  and  $\mathcal{F}_\lambda$  is the class of all circuits having input length  $k = k(\lambda)$  and depth at most  $d = d(\lambda)$ . We use the matrix  $\mathbf{G}$ , defined in Sec. 2.3, in our construction and security proof.

**Setup**( $1^\lambda, 1^\ell, \mathcal{F}_\lambda$ ): On input  $1^\lambda, 1^\ell$  and  $\mathcal{F}_\lambda$ , the setup algorithm defines the parameters  $n = n(\lambda), m = m(\lambda), q = q(\lambda)$ . It then proceeds as follows.

1. Sample  $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$  such that  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ .
2. Sample random matrices  $(\mathbf{B}_1, \dots, \mathbf{B}_k) \leftarrow (\mathbb{Z}_q^{n \times m})^k, \mathbf{D} \leftarrow \mathbb{Z}_q^{n \times \ell}$ .
3. Output the master public-key  $\text{mpk} = (\mathbf{A}, \mathbf{B}_1, \dots, \mathbf{B}_k, \mathbf{D})$  and the master secret-key  $\text{msk} = \mathbf{T}_\mathbf{A}$ . We assume that  $\text{mpk}$  also contains a set of public parameters  $\text{param} = \{n, m, q, \ell, X, V, K, \rho, \sigma, \tau\}$ .

**KeyGen**( $\text{mpk}, \text{msk}, f, \mathbf{y}$ ): The key generation algorithm takes as input  $\text{mpk}, \text{msk}$ , a function  $f \in \mathcal{F}_\lambda$  and a vector  $\mathbf{y} \in \mathcal{Y}_\lambda$ , and works as follows.

1. Compute  $\mathbf{B}_f = \text{Eval}_{\text{pk}}(f, (\mathbf{B}_1, \dots, \mathbf{B}_k))$  where  $\mathbf{B}_f \in \mathbb{Z}_q^{n \times m}$ .
2. Compute  $\mathbf{R}_f \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}_f, \mathbf{D}, \rho)$  so that  $(\mathbf{A} | \mathbf{B}_f) \cdot \mathbf{R}_f = \mathbf{D}$ .
3. Output the secret-key as  $\text{sk}_{f, \mathbf{y}} = \mathbf{R}_f \cdot \mathbf{y}$ . We assume that the secret-key trivially includes  $f$  and  $\mathbf{y}$ .

**Enc**( $\text{mpk}, \mathbf{a}, \mathbf{x}$ ): The encryption algorithm takes as input  $\text{mpk}$ , an attribute  $\mathbf{a} = (a_1, \dots, a_k) \in \{0, 1\}^k$  and a message  $\mathbf{x} \in \mathcal{X}$ . It proceeds as follows.

1. Compute  $\mathbf{H}_\mathbf{a} = (\mathbf{A} | a_1 \mathbf{G} + \mathbf{B}_1 | \dots | a_k \mathbf{G} + \mathbf{B}_k) \in \mathbb{Z}_q^{n \times m(k+1)}$ .
2. Sample  $\mathbf{s} \leftarrow \mathbb{Z}_q^n$  and  $\mathbf{e}_1 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^m), \mathbf{e}_2 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^\ell), \mathbf{e}_3 \leftarrow \mathcal{D}_\tau(\mathbb{Z}^\ell)$ , and matrices  $\mathbf{S}_i \leftarrow \{\pm 1\}^{m \times m}$  for  $i \in [k]$ .
3. Set  $\mathbf{v} = (\mathbf{I}_m | \mathbf{S}_1 | \dots | \mathbf{S}_k)^\top \cdot \mathbf{e}_1 \in \mathbb{Z}_q^{m(k+1)}$ .
4. Compute  $\text{ct}_1 = \mathbf{H}_\mathbf{a}^\top \mathbf{s} + \mathbf{v} \in \mathbb{Z}_q^{m(k+1)}, \text{ct}_2 = \mathbf{D}^\top \mathbf{s} + \mathbf{e}_2 + \mathbf{e}_3 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x} \in \mathbb{Z}_q^\ell$ .
5. Output the ciphertext  $\text{ct} = (\text{ct}_1, \text{ct}_2)$ . We assume that the ciphertext includes the attribute  $\mathbf{a}$ .

**Dec**( $\text{mpk}, \text{sk}_{f, \mathbf{y}}, \text{ct}$ ): The decryption algorithm takes as input  $\text{mpk}$ , a secret-key  $\text{sk}_{f, \mathbf{y}}$  corresponding to a function  $f$  and a predicate vector  $\mathbf{y}$  and a ciphertext  $\text{ct}$  associated with an attribute  $\mathbf{a}$ . It proceeds as follows.

1. Parse  $\text{ct} = (\text{ct}_1, \text{ct}_2)$  where  $\text{ct}_1 = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_k) \in (\mathbb{Z}_q^m)^{k+1}, \text{ct}_2 \in \mathbb{Z}_q^\ell$  and  $\text{sk}_{f, \mathbf{y}} \in \mathbb{Z}^{2m}$ .
2. Compute  $\mathbf{c}_f = \text{Eval}_{\text{ct}}(f, ((a_i, \mathbf{B}_i, \mathbf{c}_i)_{i=1}^k))$  where  $\mathbf{a} = (a_1, \dots, a_k)$ .
3. Compute  $\zeta' = \mathbf{y}^\top \text{ct}_2 - \text{sk}_{f, \mathbf{y}}^\top \cdot (\mathbf{c}_0 | \mathbf{c}_f)$ .
4. Output  $\zeta \in \{0, \dots, K\}$  which minimizes  $|\lfloor \frac{q}{K} \rfloor \cdot \zeta - \zeta'|$ .

**Correctness.** For correctness we first observe that  $\mathbf{c}_i = (a_i \mathbf{G} + \mathbf{B}_i)^\top \mathbf{s} + \mathbf{S}_i^\top \mathbf{e}_1$  with  $\|\mathbf{S}_i^\top \mathbf{e}_1\| < \sigma \sqrt{m}$  for all  $i \in [k]$ . Therefore, using Theorem 4, we have  $\mathbf{c}_f = (f(\mathbf{a}) \mathbf{G} + \mathbf{B}_f)^\top \mathbf{s} + \mathbf{e}_f \in \mathbb{Z}_q^m$  where  $\|\mathbf{e}_f\| < \sigma \sqrt{m} \cdot \gamma_{\mathcal{F}}$ . Consequently,

$$(c_0|c_f) = (\mathbf{A}|f(\mathbf{a})\mathbf{G} + \mathbf{B}_f)^\top \mathbf{s} + (e_1|e_f) \in \mathbb{Z}_q^{2m}.$$

Now, the secret-key  $\text{sk}_{f,\mathbf{y}} = \mathbf{R}_f \cdot \mathbf{y}$  where  $\mathbf{R}_f$  is sampled from  $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{D}}(\mathbf{A}|\mathbf{B}_f))$ . Thus,  $(\mathbf{A}|\mathbf{B}_f) \cdot \mathbf{R}_f = \mathbf{D}$  and  $\|\mathbf{R}_f\| < \rho\sqrt{2m\ell}$ . Since  $e_2 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^\ell)$ ,  $e_3 \leftarrow \mathcal{D}_\tau(\mathbb{Z}^\ell)$ , with overwhelming probability we have  $\|e_2\| < \sigma\sqrt{\ell}$  and  $\|e_3\| < \tau\sqrt{\ell}$ . Finally, if  $f(\mathbf{a}) = 0$  then the element  $\zeta'$  can be viewed as

$$\begin{aligned} \zeta' &= \mathbf{y}^\top \text{ct}_2 - \text{sk}_{f,\mathbf{y}}^\top \cdot (c_0|c_f) \\ &= \mathbf{y}^\top (\mathbf{D}^\top \mathbf{s} + e_2 + e_3 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}) - (\mathbf{R}_f \cdot \mathbf{y})^\top \cdot ((\mathbf{A}|\mathbf{B}_f)^\top \mathbf{s} + (e_1|e_f)) \\ &= \lfloor \frac{q}{K} \rfloor \cdot \langle \mathbf{x}, \mathbf{y} \rangle + \mathbf{y}^\top (e_2 + e_3) - (\mathbf{R}_f \mathbf{y})^\top (e_1|e_f) = \lfloor \frac{q}{K} \rfloor \cdot \langle \mathbf{x}, \mathbf{y} \rangle + \text{error} \end{aligned}$$

and  $|\text{error}| < V\ell(\sigma + \tau) + 2\rho\sigma V\ell m(1 + \gamma_{\mathcal{F}})$  with overwhelming probability. To ensure the correct decryption we need to set  $q > 4KV\ell(\sigma + \tau) + 8\rho\sigma KV\ell m(1 + \gamma_{\mathcal{F}})$  so that  $\zeta = \langle \mathbf{x}, \mathbf{y} \rangle$  minimizes  $|\lfloor \frac{q}{K} \rfloor \cdot \zeta - \zeta'|$ .

**Theorem 6 (1-bounded coSel-IND security)** *Assuming the modified variant of ALS-IPFE scheme of Sec. 3 with parameters  $n, q, m, \sigma, \rho, \alpha$  is secure under  $\text{LWE}_{q,\alpha,n}$  and the parameters additionally satisfy  $m \geq 6n \log q$ ,  $q > 4KV\ell(\sigma + \tau) + 8\rho\sigma KV\ell m(1 + \gamma_{\mathcal{F}})$ , the above ABIPFE scheme with  $\tau > 2C\rho\sigma(2\sqrt{m} + \sqrt{\ell})\gamma_{\mathcal{F}}$  for a constant  $C$  is 1-bounded coSel-IND secure under the  $\text{LWE}_{q,\alpha,n}$  assumption.*

*Proof.* The proof is done by considering the sequence of games used in the selectively secure ABE of [14]. We also incorporate the idea of [3] to simulate the secret-key queries correspond to the target accepting function. However, we make crucial changes along the way to let proof go through. As in Def. 2 with  $Q = 1$ , we assume that the adversary  $\mathcal{A}$  submits a target attribute  $\mathbf{a}^*$  and a target accepting function  $f^*$  (i.e.  $f^*(\mathbf{a}^*) = 0$ ) before seeing the master public-key. A secret-key query  $(f, \mathbf{y})$  should satisfy either  $f(\mathbf{a}^*) = 1$  or  $(f = f^* \wedge \langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle)$  where  $\mathbf{x}_0, \mathbf{x}_1$  are the challenge messages chosen adaptively from  $\mathcal{X}_\lambda$ .

**Game 0:** This is the standard ABIPFE experiment as defined in Def. 2.

**Game 1:** We modify the setup algorithm. The challenger selects a random matrix  $\mathbf{A}$  distributed uniformly over  $\mathbb{Z}_q^{n \times m}$ , instead of sampling  $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$ . However, a short basis of  $\Lambda_q^\perp(\mathbf{A})$  is required to answer  $\mathcal{A}$ 's secret-key queries. For that, we may enumerate all short bases of  $\Lambda_q^\perp(\mathbf{A})$  and select one of these bases as  $\mathbf{T}_\mathbf{A}$ . Note that, from Lemma 2, we have  $\min\|\tilde{\mathbf{B}}\| < O(m)$  where minimum is taken over all ordered bases of  $\Lambda_q^\perp(\mathbf{A})$ . To apply  $\text{SampleD}$  with the input basis  $\mathbf{B}$  (Theorem 3), we need to set  $\rho > \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log m})$ . Since  $m = \Theta(n \log q)$ , this suggests to set  $\rho > n \cdot \omega(\sqrt{n})$ .

The challenger is inefficient in this game, but this should not be a problem as long as we establish statistical indistinguishability between the games. The matrix  $\mathbf{A}$  used in game 0 is generated by  $\text{TrapGen}(1^n, 1^m, q)$  and Lemma 3 states that the distribution of  $\mathbf{A}$  is statistically close to uniform over  $\mathbb{Z}_q^{n \times m}$ . Therefore, game 0 and game 1 are statistically indistinguishable as required. Such basis

selection process followed by a statistical argument has been used in [3] to simulate the key queries for an adversary.

**Game 2:** In this game the public matrix  $\mathbf{D} \in \mathbb{Z}_q^{n \times \ell}$  is programmed by the challenger as follows. First, it samples  $\mathbf{Z}_1, \mathbf{Z}_2 \leftarrow \mathcal{D}_\rho(\mathbb{Z}^{m \times \ell})$  and set  $\mathbf{D}_1 = \mathbf{A}\mathbf{Z}_1$ . Since  $\mathcal{A}$  submits the target accepting function  $f^*$  before setup, the challenger computes  $\mathbf{B}_{f^*} \leftarrow \text{Eval}_{\text{pk}}(f^*, (\mathbf{B}_1, \dots, \mathbf{B}_k))$  and set  $\mathbf{D} = \mathbf{D}_1 + \mathbf{B}_{f^*}\mathbf{Z}_2$ . In particular, if we take  $\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \in \mathbb{Z}^{2m \times \ell}$ , then  $\mathbf{D} = (\mathbf{A}|\mathbf{B}_{f^*})\mathbf{Z}$ . Instead of computing  $\mathbf{R}_{f^*} \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}_{f^*}, \mathbf{D}, \rho)$ , the challenger uses  $\mathbf{Z}$  and answers secret-key queries for  $(f^*, \mathbf{y})$  as  $\text{sk}_{f^*, \mathbf{y}} = \mathbf{Z} \cdot \mathbf{y}$ . Note that, both  $\mathbf{R}_{f^*}$  and  $\mathbf{Z}$  follow the same distribution  $\mathcal{D}_\rho(\mathbb{Z}^{2m \times \ell})$ , as given in Lemma 5. However, the challenger still computes  $\mathbf{R}_f \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}_f, \mathbf{D}, \rho)$  and outputs  $\mathbf{R}_f \cdot \mathbf{y}$  as a reply to a secret-key query corresponding to  $(f, \mathbf{y})$  if  $f(\mathbf{a}^*) = 1$ .

We show that  $\mathbf{D}$  is uniformly distributed over  $\mathbb{Z}_q^{n \times \ell}$ . Specifically, we observe that for a matrix  $\mathbf{A}$  uniform over  $\mathbb{Z}_q^{n \times m}$  and a short basis  $\mathbf{T}_\mathbf{A}$  the distributions

$$\text{Dist}_1 := (\mathbf{A}, \mathbf{Z}_1, \mathbf{D}_1) \text{ s.t. } \mathbf{D}_1 \leftarrow \mathbb{Z}^{n \times \ell}, \mathbf{Z}_1 \leftarrow \text{SamplePre}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{D}_1, \rho),$$

$$\text{Dist}_2 := (\mathbf{A}, \mathbf{Z}_1, \mathbf{A}\mathbf{Z}_1) \text{ s.t. } \mathbf{Z}_1 \leftarrow \mathcal{D}_\rho(\mathbb{Z}^{m \times \ell})$$

are statistically close by Lemma 3. Therefore,  $\mathbf{D}_1 = \mathbf{A}\mathbf{Z}_1$  is statistically close to uniform over  $\mathbb{Z}_q^{n \times \ell}$  and hence the matrix  $\mathbf{D} = \mathbf{D}_1 + \mathbf{B}_{f^*}\mathbf{Z}_2$  of game 2 is also statistically close to uniform over  $\mathbb{Z}_q^{n \times \ell}$ . Thus, game 1 and game 2 are statistically indistinguishable.

**Game 3:** Instead of selecting  $(\mathbf{B}_1, \dots, \mathbf{B}_k)$  uniformly from  $(\mathbb{Z}_q^{n \times m})^k$ , the challenger first chooses random matrices  $\mathbf{S}_i^* \leftarrow \{\pm 1\}^{m \times m}$  in advance and uses the challenge attribute  $\mathbf{a}^* = (a_1^*, \dots, a_k^*)$  to set  $\mathbf{B}_i = \mathbf{A}\mathbf{S}_i^* + a_i^*\mathbf{G}$  for all  $i \in [k]$ . Note that, the matrices  $\mathbf{S}_1^*, \dots, \mathbf{S}_k^*$  will be utilized to create the challenge ciphertext  $\text{ct}^* = (\text{ct}_1^*, \text{ct}_2^*)$ . In particular, a fixed  $\mathbf{e}_1 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^m)$  and low-norm vectors  $\mathbf{S}_i^* \cdot \mathbf{e}_1 \in \mathbb{Z}^m$  for all  $i \in [k]$  are used to create  $\text{ct}_1^*$ .

Observe that the distribution  $(\mathbf{A}, \mathbf{A}\mathbf{S}_i^*, \mathbf{S}_i^*\mathbf{e}_1)$  is statistically close to the distribution  $(\mathbf{A}, \mathbf{B}', \mathbf{S}_i^*\mathbf{e}_1)$  by left-over hash lemma (Lemma 6) where  $\mathbf{B}'$  is uniform over  $\mathbb{Z}_q^{n \times m}$ . This holds for all  $i \in [k]$  and hence all matrices  $\mathbf{A}\mathbf{S}_i^*$  are statistically close to uniform over  $\mathbb{Z}_q^{n \times m}$ . In other words, given  $(\mathbf{S}_1^* | \dots | \mathbf{S}_k^*) \cdot \mathbf{e}_1$ , all matrices  $\mathbf{B}_i = \mathbf{A}\mathbf{S}_i^* + a_i^*\mathbf{G}$  of game 3 are statistically close to uniform as in game 2. Thus, game 2 and game 3 are statistically indistinguishable.

**Game 4:** In this game, we make the challenger efficient, that is the short basis  $\mathbf{T}_\mathbf{A}$  is not required in the key query phase. Recall that a secret-key query  $(f, \mathbf{y})$  of  $\mathcal{A}$  should satisfy either  $f(\mathbf{a}^*) = 1$  or  $(f = f^* \wedge \langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle)$ . If  $f = f^*$ , the challenger uses the secret matrix  $\mathbf{Z}$  to send the secret-key as  $\text{sk}_{f^*, \mathbf{y}} = \mathbf{Z} \cdot \mathbf{y}$  as in the previous game. When  $f(\mathbf{a}^*) = 1$ , instead of sampling  $\mathbf{R}_f \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}_f, \mathbf{D}, \rho)$  satisfying,  $(\mathbf{A}|\mathbf{B}_f)\mathbf{R}_f = \mathbf{D}$  the challenger does the following.

1. Compute  $\mathbf{S}_f = \text{Eval}_{\text{sim}}(f, ((a_i^*, \mathbf{S}_i^*))_{i=1}^k, \mathbf{A})$  which satisfies  $\mathbf{A}\mathbf{S}_f + \mathbf{G} = \mathbf{B}_f$  and  $\|\mathbf{S}_f\|_2 < \gamma_{\mathcal{F}}$  by Theorem 4.

2. Sample  $\mathbf{R}_f \leftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{S}_f, 1, \mathbf{D}, \rho)$  which is distributed according to  $\mathcal{D}_\rho(\Lambda_q^{\mathbf{D}}(\mathbf{A}|\mathbf{A}\mathbf{S}_f + \mathbf{G}))$  by Lemma 5.
3. Finally, the challenger outputs  $\text{sk}_{f,\mathbf{y}} = \mathbf{R}_f \cdot \mathbf{y}$

Observe that  $\mathbf{R}_f$  satisfies  $(\mathbf{A}|\mathbf{A}\mathbf{S}_f + \mathbf{G})\mathbf{R}_f = (\mathbf{A}|\mathbf{B}_f)\mathbf{R}_f = \mathbf{D}$  as required. To apply **SampleLeft**, we need to set  $\rho \geq \sqrt{5} \cdot (1 + \|\mathbf{S}_f\|_2) \cdot \omega(\sqrt{\log m})$ . We also require  $\rho > n \cdot \omega(\sqrt{n})$  as suggested in game 2. Combining, we set  $\rho > n\gamma_{\mathcal{F}} \cdot \omega(\sqrt{n})$ . The public parameters and secret-key queries in this game are statistically close to that of game 3. Hence,  $\mathcal{A}$ 's advantage in distinguishing between game 3 and game 4 is at most negligible in  $\lambda$ .

**Game 5:** In this game, we rely on the security of ALS-IPFE described in Sec. 3 to establish the indistinguishability of the challenge ciphertext encrypting  $\mathbf{x}_b$  for  $b \leftarrow \{0, 1\}$ . We consider an intermediate adversary  $\mathcal{B}$  that interacts with the ALS-IPFE challenger. Let  $\mathcal{B}$  receives the master public-key  $\text{mpk}_{\text{ALS}} = (\mathbf{A}_{\text{ALS}}, \mathbf{D}_{\text{ALS}})$  from the ALS-IPFE challenger and a pair of attribute and target accepting function  $(\mathbf{a}^*, f^*)$  from  $\mathcal{A}$ . Now,  $\mathcal{B}$  simulates  $\mathcal{A}$  as follows.

$\mathcal{B}(1^\lambda, \text{mpk}_{\text{ALS}}, \mathbf{a}^*, f^*)$ :

**Setup.** Pick  $\mathbf{Z}_2 \leftarrow \mathcal{D}_\rho(\mathbb{Z}^{m \times \ell})$  and  $\mathbf{S}_i^* \leftarrow \{\pm 1\}^{m \times m}$  for  $i \in [k]$ , and set

$$\mathbf{A} = \mathbf{A}_{\text{ALS}}^\top, \quad \mathbf{B}_i = \mathbf{A}\mathbf{S}_i^* - a_i^*\mathbf{G} \quad \forall i \in [k], \quad \mathbf{D} = \mathbf{D}_{\text{ALS}}^\top + \mathbf{B}_{f^*}\mathbf{Z}_2,$$

where  $\mathbf{a}^* = (a_1^*, \dots, a_k^*)$  and  $\mathbf{B}_{f^*} = \text{Eval}_{\text{pk}}(f^*, (\mathbf{B}_1, \dots, \mathbf{B}_k))$ . It sends the master public-key as  $\text{mpk} = (\mathbf{A}, \mathbf{B}_1, \dots, \mathbf{B}_k, \mathbf{D})$ .

**Secret-key queries.** Suppose  $\mathcal{A}$  asks a secret-key for a tuple  $(f, \mathbf{y})$ .

- (a) If  $f = f^*$  then  $\mathcal{B}$  requests a secret-key for  $\mathbf{y}$  from the ALS-IPFE challenger.

Let  $\text{sk}_{\mathbf{y}}^{\text{ALS}}$  be the secret-key. Then  $\mathcal{B}$  sends  $\text{sk}_{f^*,\mathbf{y}} = \begin{pmatrix} (\text{sk}_{\mathbf{y}}^{\text{ALS}})^\top \\ \mathbf{Z}_2\mathbf{y} \end{pmatrix}$

as the secret-key for  $(f^*, \mathbf{y})$ .

- (b) If  $f(\mathbf{a}^*) = 1$  then  $\mathcal{B}$  uses  $\text{Eval}_{\text{sim}}$  and **SampleLeft** to obtain a matrix  $\mathbf{R}_f \in \mathbb{Z}^{2m \times \ell}$  and outputs  $\mathbf{R}_f \cdot \mathbf{y}$  as in the previous game.

**Challenge ciphertext.** Let  $(\mathbf{x}_0, \mathbf{x}_1)$  be the challenge messages submitted by  $\mathcal{A}$ . Then,  $\mathcal{B}$  submits the same to the ALS-IPFE challenger and receives  $\text{ct}_b^{\text{ALS}} = (\text{ct}_1^{\text{ALS}}, \text{ct}_2^{\text{ALS}})$ . Now,  $\mathcal{B}$  computes and sends the challenge ciphertext  $\text{ct}^* = (\text{ct}_1^*, \text{ct}_2^*)$  for  $\mathcal{A}$  as

$$\text{ct}_1^* = \text{ct}_1^{\text{ALS}} + (\mathbf{S}^*)^\top \cdot \text{ct}_1^{\text{ALS}} \quad \text{and} \quad \text{ct}_2^* = \text{ct}_2^{\text{ALS}} + \mathbf{Z}_2^\top \cdot \mathbf{c}_{f^*} + \text{NoiseGen}(\mathbf{Z}_2^\top, s)$$

where we take  $\mathbf{S}^* = (\mathbf{S}_1^* | \dots | \mathbf{S}_k^*) \in \{\pm 1\}^{m \times km}$ ,  $\text{ct}_1^* = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_k) \in (\mathbb{Z}_q^m)^{k+1}$ ,  $\mathbf{c}_{f^*} = \text{Eval}_{\text{ct}}(f^*, ((a_i^*, \mathbf{B}_i, \mathbf{c}_i))_{i=1}^k)$  and  $\text{NoiseGen}$  is the randomized algorithm with  $s > s_1(\mathbf{Z}_2^\top)$  from Lemma 7.

We show that the distribution of the master public-key, secret-key queries and the challenge ciphertext are statistically close to that of in game 4. Let  $\mathbf{D}_{\text{ALS}} = \mathbf{Z}_{\text{ALS}}\mathbf{A}_{\text{ALS}}$  for some matrix  $\mathbf{Z}_{\text{ALS}} \leftarrow \mathcal{D}_\rho(\mathbb{Z}^{\ell \times m})$ . Therefore, we have

$$\mathbf{D} = (\mathbf{Z}_{\text{ALS}}\mathbf{A}_{\text{ALS}})^\top + \mathbf{B}_{f^*}\mathbf{Z}_2 = \mathbf{A}\mathbf{Z}_{\text{ALS}}^\top + \mathbf{B}_{f^*}\mathbf{Z}_2 = (\mathbf{A}|\mathbf{B}_{f^*})\mathbf{Z}$$



where  $\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_{\text{ALS}}^\top \\ \mathbf{Z}_2 \end{pmatrix}$  is distributed according to  $\mathcal{D}_\rho(\mathbb{Z}^{2m \times \ell})$ . Note that,  $\mathbf{Z}_{\text{ALS}}$  plays the role of master secret-key of ALS-IPFE and the secret-keys of the form  $\text{sk}_{f^*, \mathbf{y}} = \begin{pmatrix} \mathbf{Z}_{\text{ALS}}^\top \mathbf{y} \\ \mathbf{Z}_2 \mathbf{y} \end{pmatrix} = \mathbf{Z} \cdot \mathbf{y}$  are distributed similar to the previous game. Thus, the master public-key  $\text{mpk}$  and the secret-keys  $\text{sk}_{f^*, \mathbf{y}}$  for  $(f^*, \mathbf{y})$  are distributed according to game 4. Moreover, secret-keys for  $(f, \mathbf{y})$  satisfying  $f(\mathbf{a}^*) = 1$  are identically distributed as in game 4.

Now, let  $\text{ct}_1^{\text{ALS}} = \mathbf{A}_{\text{ALS}} \mathbf{s} + \mathbf{e}_1$  and  $\text{ct}_2^{\text{ALS}} = \mathbf{D}_{\text{ALS}} \mathbf{s} + \mathbf{e}_2 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}_b$  for some  $\mathbf{e}_1 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^m)$  and  $\mathbf{e}_2 \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^\ell)$ . Hence, we can write the challenge ciphertext

$$\begin{aligned} \text{ct}_1^* &= \text{ct}_1^{\text{ALS}} + (\mathbf{S}^*)^\top \cdot \text{ct}_1^{\text{ALS}} = \mathbf{A}^\top \mathbf{s} + \mathbf{e}_1 + (\mathbf{S}^*)^\top \cdot (\mathbf{A}^\top \mathbf{s} + \mathbf{e}_1) \\ &= (\mathbf{A} | \mathbf{A} \mathbf{S}^*)^\top \mathbf{s} + (\mathbf{I}_m | \mathbf{S}^*)^\top \cdot \mathbf{e}_1 = \mathbf{H}_{\mathbf{a}^*}^\top \mathbf{s} + \mathbf{v} \end{aligned}$$

where  $\mathbf{H}_{\mathbf{a}^*} = (\mathbf{A} | a_1^* \mathbf{G} + \mathbf{B}_1 | \dots | a_k^* \mathbf{G} + \mathbf{B}_k) = (\mathbf{A} | \mathbf{A} \mathbf{S}^*)$  and  $\mathbf{v} = (\mathbf{I}_m | \mathbf{S}^*)^\top \cdot \mathbf{e}_1$ . Observe that, by Theorem 4,  $\text{Eval}_{\text{ct}}(f^*, ((a_i^*, \mathbf{B}_i, \mathbf{c}_i))_{i=1}^k) = (f^*(\mathbf{a}^*) \mathbf{G} + \mathbf{B}_{f^*})^\top \mathbf{s} + \mathbf{e}_{f^*} = \mathbf{B}_{f^*}^\top \mathbf{s} + \mathbf{e}_{f^*} = \mathbf{c}_{f^*}$  with  $\|\mathbf{e}_{f^*}\| < \sigma \sqrt{m} \cdot \gamma_{\mathcal{F}}$  which implies

$$\begin{aligned} \text{ct}_2^* &= \text{ct}_2^{\text{ALS}} + \mathbf{Z}_2^\top \cdot \mathbf{c}_{f^*} + \text{NoiseGen}(\mathbf{Z}_2^\top, s) \\ &= (\mathbf{D} - \mathbf{B}_{f^*} \mathbf{Z}_2)^\top \mathbf{s} + \mathbf{e}_2 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}_b + \mathbf{Z}_2^\top \cdot (\mathbf{B}_{f^*}^\top \mathbf{s} + \mathbf{e}_{f^*}) + \text{NoiseGen}(\mathbf{Z}_2^\top, s) \\ &= \mathbf{D}^\top \mathbf{s} + \mathbf{e}_2 + \mathbf{Z}_2^\top \mathbf{e}_{f^*} + \text{NoiseGen}(\mathbf{Z}_2^\top, s) + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}_b \end{aligned}$$

From Lemma 1, we have  $s_1(\mathbf{Z}_2^\top) \leq C\rho(2\sqrt{m} + \sqrt{\ell})$  and Lemma 7 implies that  $\mathbf{Z}_2^\top \mathbf{e}_{f^*} + \text{NoiseGen}(\mathbf{Z}_2^\top, s)$  is distributed statistically close to  $\mathcal{D}_\tau(\mathbb{Z}^\ell)$  where  $\tau > 2C\rho\sigma(2\sqrt{m} + \sqrt{\ell})\gamma_{\mathcal{F}}$ . Therefore, we can write  $\text{ct}_2^* = \mathbf{D}^\top \mathbf{s} + \mathbf{e}_2 + \mathbf{e}_3 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}_b$  where  $\mathbf{e}_2, \mathbf{e}_3$  are distributed as  $\mathcal{D}_\sigma(\mathbb{Z}^\ell)$  and  $\mathcal{D}_\tau(\mathbb{Z}^\ell)$  respectively. This proves that the challenge ciphertext is distributed statistically close to that of in the previous game. Also, the advantage of  $\mathcal{A}$  in guessing the challenge bit is upper bounded by the advantage of  $\mathcal{B}$  in breaking the security of ALS-IPFE scheme.

**Parameter Setting.** First we choose  $n, m, q, \sigma, \rho$  as in ALS-IPFE of Sec. 3. We modify them step by step according to our requirement for correctness and security of our ABIPFE scheme. The modifications are made without violating the security of ALS-IPFE.

1. For `TrapGen` algorithm we set  $m \geq 6n \log q$ .
2. To obtain a short basis  $\mathbf{T}_{\mathbf{A}}$  for a uniformly chosen matrix  $\mathbf{A}$  as required in game 1, we set  $\rho > n \cdot \omega(\sqrt{n})$ .
3. The parameters already satisfy the constrain in the left-over hash lemma (game 3 of the security proof).
4. For `SampleRight` and `SampleLeft` we need to set  $\rho > \max\{\|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \cdot \omega(\sqrt{\log m}), \sqrt{5}(1 + \|\mathbf{S}_f\|_2)\omega(\sqrt{\log m})\}$  where  $\|\mathbf{S}_f\|_2 < \gamma_{\mathcal{F}}$ . This is due to correctness and game 4 of the security proof. Thus, combining with step 2, we can set  $\rho > n\gamma_{\mathcal{F}} \cdot \omega(\sqrt{n})$ .

5. To apply NoiseGen in game 5, we need to keep  $\tau > 2C\rho\sigma(2\sqrt{m} + \sqrt{\ell})\gamma_{\mathcal{F}}$ .
6. For the hardness of  $\text{LWE}_{q,\alpha,n}$  we want the standard deviation to satisfy  $\alpha q > 2\sqrt{n}$ .

Finally, the parameters of our ABIPFE can be set as

$$q > 4KV\ell(\sigma + \tau) + 8\rho\sigma KV\ell m(1 + \gamma_{\mathcal{F}}), \quad \sigma = 2C'\alpha q(\sqrt{m} + \sqrt{n} + \sqrt{\ell})$$

$$m \geq 6n \log q, \quad \rho > n\gamma_{\mathcal{F}} \cdot \omega(\sqrt{n}), \quad \tau > 2C\rho\sigma(2\sqrt{m} + \sqrt{\ell})\gamma_{\mathcal{F}}$$

where  $C$  (as in game 5),  $C'$  (as in ALS-IPFE) are constants.

## 5 Attribute-based Multi-input IPFE from LWE

We define an  $\text{ABMIPFE}_{n,m}$  scheme with access control given by a class of polynomial size circuits where  $n$  denotes the number of encryption slots and  $m$  denotes the number of attributes supported by each slot. Consider a class of attributes  $\text{Att} = \{((\mathbf{a}_1^{(j)}, \dots, \mathbf{a}_n^{(j)}))_{j=1}^m\}$  where  $i$ -th encryption slot is associated to the attribute set  $\text{Att}_i = \{\mathbf{a}_i^{(1)}, \dots, \mathbf{a}_i^{(m)}\}$  and  $\mathbf{a}_i^{(j)} \in \{0, 1\}^k$  for all  $i \in [n], j \in [m]$ . We represent the attribute class as  $\text{Att} = [\text{Att}_1 | \dots | \text{Att}_n]$ . The  $i$ -th encryption slot encrypts a vector  $\mathbf{x} \in \mathbb{Z}_q^\ell$  with respect to an attribute  $\mathbf{a}_i^{(j)}$  for  $j \in [m]$ . We denote  $\mathcal{F}_\lambda^{d,k}$  by the set of all polynomial size circuits with input space  $\{0, 1\}^k$  and depth bounded by  $d$ . A secret-key is generated for a tuple  $(S \subseteq [n], (f_i, \mathbf{y}_i)_{i \in S})$  where  $f_i \in \mathcal{F}_\lambda^{d,k}$ ,  $\mathbf{y}_i \in \mathbb{Z}_q^\ell$  for all  $i \in S$ . The secret-key allows a receiver to learn  $\sum_{i \in S} \langle \mathbf{x}_i, \mathbf{y}_i \rangle$  if  $f_i(\mathbf{a}_i^{(j)}) = 0$  for all  $i \in S$  where  $\mathbf{x}_i$  is encrypted for the  $i$ -th slot with an attribute  $\mathbf{a}_i^{(j)} \in \text{Att}_i$ . For security, we first consider adaptive indistinguishability (**Adp-IND**) where the adversary  $\mathcal{A}$  has the freedom to choose secret-key queries and encryption queries depending on the  $\text{mpk}$ . We also define a weaker security notion called  $Q$ -bounded co-adaptive indistinguishability (**coAdp-IND**) where  $\mathcal{A}$  is restricted to submit all functions  $f_1, \dots, f_Q$  to be queried along with the predicate vectors in the key query phase before seeing the  $\text{mpk}$ . This is similar to the **coSel-IND** notion of ABIPFE. We now formally define  $\text{ABMIPFE}_{n,m}$  and its security notions.

An  $\text{ABMIPFE}_{n,m}$  for a class of functions  $\mathcal{F}_\lambda^{d,k}$ , a class of attributes  $\text{Att}$ , a predicate space  $\mathcal{Y}_\lambda$  and message space  $\mathcal{X}_\lambda$  consists of four PPT algorithms  $\text{ABMIPFE}_{n,m} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$  working as follows:

- $(\text{mpk}, \text{msk}, \{\text{ek}_i\}_{i=1}^n) \leftarrow \text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda^{d,k}, \text{Att})$  : The setup on inputs a security parameter  $\lambda$ , a vector length  $\ell$ , a function class  $\mathcal{F}_\lambda^{d,k}$  and a class of attributes  $\text{Att}$ , outputs a master public-key  $\text{mpk}$ , a master secret-key  $\text{msk}$  and  $n$  encryption keys  $\text{ek}_1, \dots, \text{ek}_n$ .
- $\text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{msk}, S, (f_i, \mathbf{y}_i)_{i \in S})$  : The key generation algorithm on input the master secret-key  $\text{msk}$ , a set  $S \subseteq [n]$  and function-predicate pairs  $(f_i \in \mathcal{F}_\lambda^{d,k}, \mathbf{y}_i \in \mathcal{Y}_\lambda)$  for  $i \in S$ , outputs a secret-key  $(S, \text{sk}_{f,\mathbf{y}})$ .

- $\text{ct} \leftarrow \text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}, \mathbf{x})$  : The encryption algorithm on input the master public-key  $\text{mpk}$ , an encryption key  $\text{ek}_i$ , an attribute  $\mathbf{a} \in \text{Att}_i$  and a message vector  $\mathbf{x} \in \mathcal{X}_\lambda$ , outputs a ciphertext  $\text{ct}$ .
- $\perp$  or  $\zeta \leftarrow \text{Dec}(\text{mpk}, \text{S}, \text{sk}_{f,\mathbf{y}}, \{\text{ct}_i\}_{i=1}^n)$  : The decryption algorithm takes as input the master public-key  $\text{mpk}$ , a decryption key  $(\text{S}, \text{sk}_{f,\mathbf{y}})$  and  $n$  ciphertexts  $\text{ct}_1, \dots, \text{ct}_n$ , and outputs a message  $\zeta$  or  $\perp$  indicating failure.

**Definition 3 (Correctness)** An  $\text{ABMIPFE}_{n,m}$  scheme for a class of functions  $\mathcal{F}_\lambda^{d,k}$  and a class of attributes  $\text{Att} = [\text{Att}_1 | \dots | \text{Att}_n]$  is said to be correct if for any  $\lambda \in \mathbb{N}$ ,  $\text{S} \subseteq [n]$ ,  $(f_i \in \mathcal{F}_\lambda^{d,k}, \mathbf{y}_i \in \mathcal{Y}_\lambda)$  for  $i \in \text{S}$  and  $(\mathbf{a}_i \in \text{Att}_i, \mathbf{x}_i \in \mathcal{X}_\lambda)$  for  $i \in [n]$ , the following

$$\Pr \left[ \begin{array}{l} \zeta = \sum_{i \in \text{S}} \langle \mathbf{x}_i, \mathbf{y}_i \rangle \wedge \\ f_i(\mathbf{a}_i) = 0 \quad \forall i \in \text{S} \end{array} : \begin{array}{l} (\text{mpk}, \text{msk}, \{\text{ek}_i\}_{i=1}^n) \leftarrow \text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda^{d,k}, \text{Att}), \\ \text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{msk}, \text{S}, (f_i, \mathbf{y}_i)_{i \in \text{S}}), \\ \text{ct}_i \leftarrow \text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i, \mathbf{x}_i) \quad \forall i \in [n], \\ \zeta \leftarrow \text{Dec}(\text{mpk}, \text{S}, \text{sk}_{f,\mathbf{y}}, \text{ct}_1, \dots, \text{ct}_n) \end{array} \right]$$

is  $1 - \text{negl}(\lambda)$  for some negligible function  $\text{negl}$ .

**Definition 4 (Adp-IND security for  $\text{ABMIPFE}_{n,m}$ )** For a  $\text{ABMIPFE}_{n,m}$  scheme for class of functions  $\mathcal{F}_\lambda^{d,k}$ , a class of attributes  $\text{Att} = [\text{Att}_1 | \dots | \text{Att}_n]$ , a predicate space  $\mathcal{Y}_\lambda$ , message space  $\mathcal{X}_\lambda$  and for any PPT adversary  $\mathcal{A}$ , we define Adp-IND security experiment  $\text{Expt}_{\text{ABMIPFE}_{n,m}, \mathcal{A}}^{\text{Adp-IND}}(1^\lambda, \beta)$  as follows.

1. **Setup Phase.** The challenger computes  $(\text{mpk}, \text{msk}, \{\text{ek}_i\}_{i=1}^n) \leftarrow \text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda^{d,k}, \text{Att})$  and sends  $\text{mpk}$  to  $\mathcal{A}$ .
2. **Query Phase.** During the experiment  $\mathcal{A}$  can adaptively make the following queries in any arbitrary order.
  - (a) **Corrupt Queries.**  $\mathcal{A}$  is given access to an oracle  $\mathcal{O}_{\text{Corr}}(\cdot)$  which on input  $i \in [n]$  returns  $\text{ek}_i$ . Let  $S_{\text{Corr}}$  be the set of  $i \in [n]$  queried by  $\mathcal{A}$ .
  - (b) **Key Queries.**  $\mathcal{A}$  is given access to a key generation oracle  $\mathcal{O}_{\text{KG}}(\cdot, \cdot)$  which on input  $(\text{S} \subseteq [n], (f_i \in \mathcal{F}_\lambda^{d,k}, \mathbf{y}_i \in \mathcal{Y}_\lambda)_{i \in \text{S}})$  outputs  $\text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{msk}, \text{S}, (f_i, \mathbf{y}_i)_{i \in \text{S}})$ .
  - (c) **Encryption Queries.**  $\mathcal{A}$  can query  $(i \in [n], \mathbf{a}_i^{(j)} \in \text{Att}_i, (\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}) \in \mathcal{X}_\lambda^2)$  to an encryption oracle  $\mathcal{O}_{\text{Enc}}(\cdot, \cdot, \cdot)$  which returns  $\text{ct}_i^{j,\beta} \leftarrow \text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i^{(j)}, \mathbf{x}_i^{j,\beta})$ . When  $m = 1$  and  $\text{Att} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$ , then  $\mathbf{a}_i$  is fixed in the  $i$ -th slot and  $\mathcal{A}$  only submits  $(i, (\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}))$  to the oracle. Let  $Q_i$  denote the set of queries made by  $\mathcal{A}$  for each  $i \in [n]$ .

Without loss of generality, we assume that  $\mathcal{A}$  queries to  $\mathcal{O}_{\text{Enc}}(\cdot, \cdot, \cdot)$  for all  $i \in [n]$ . Moreover, all queries  $(i, \mathbf{a}_i^{(j)}, (\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}))$  should satisfy that  $\mathbf{x}_i^{j,0} = \mathbf{x}_i^{j,1}$  if  $i \in S_{\text{Corr}}$  and  $\sum_{i \in \text{S}} \langle \mathbf{x}_i^{j,0}, \mathbf{y}_i \rangle = \sum_{i \in \text{S}} \langle \mathbf{x}_i^{j,1}, \mathbf{y}_i \rangle$  for all queries  $(\text{S}, (f_i, \mathbf{y}_i)_{i \in \text{S}})$  made to  $\mathcal{O}_{\text{KG}}(\cdot, \cdot)$ . To avoid trivial leakage due to the inner product functionality, we also require all the queries to satisfy  $\sum_{i \in \text{S}} \langle \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}, \mathbf{y}_i \rangle = \sum_{i \in \text{S}} \langle \mathbf{x}_i^{j,1} - \mathbf{x}_i^{1,1}, \mathbf{y}_i \rangle$  for all  $j$  associated to  $Q_i$ .

3. **Guess Phase.** Finally,  $\mathcal{A}$  outputs a bit  $\beta'$ . The experiment outputs 1 if  $\beta = \beta'$ .

The  $\text{ABMIPFE}_{n,m}$  is said to satisfy Adp-IND security (or simple adaptive security) if the advantage

$$\text{Adv}_{\text{ABMIPFE}_{n,m},\mathcal{A}}^{\text{Adp-IND}}(\lambda) = \left| \Pr[\text{Expt}_{\text{ABMIPFE}_{n,m},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 0) = 1] - \Pr[\text{Expt}_{\text{ABMIPFE}_{n,m},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 1) = 1] \right|$$

of  $\mathcal{A}$  in the above game is negligible in  $\lambda$ .

We consider the following weaker version of the security for  $\text{ABMIPFE}_{n,m}$ .

**Definition 5 ( $Q$ -bounded coAdp-IND security for  $\text{ABMIPFE}_{n,m}$ )** We define  $Q$ -bounded coAdp-IND security game similarly to the Adp-IND security game except that the adversary submits all the functions  $f_1, f_2, \dots, f_Q$  to be queried in the key query phase along with adaptively chosen predicate vectors before receiving the master public-key  $\text{mpk}$ . We define the advantage  $\text{Adv}_{\text{ABMIPFE}_{n,m},\mathcal{A}}^{\text{coAdp-IND}}(\lambda)$  accordingly and the  $\text{ABIPFE}_{n,m}$  is said to satisfy  $Q$ -bounded coAdp-IND security (or simply  $Q$ -bounded co-adaptive security) if the advantage is negligible in  $\lambda$ .

### 5.1 Generic Construction of ABMIPFE from ABIPFE

We utilize the transformation of [2,3] to convert a single input ABIPFE into an  $\text{ABMIPFE}_{n,1}$ . Let us consider an ABIPFE for the function class  $\mathcal{F}_\lambda^{k,d}$  along with the predicates space  $\{0, \dots, V(\lambda) - 1\}^\ell$  and message space  $\{0, \dots, X(\lambda) - 1\}^\ell$ . Combining we say that the ABIPFE is associated with a class  $(\mathcal{F}_\lambda^{k,d}, \mathcal{F}_\lambda^{\ell,V,X})$ . We construct an  $\text{ABMIPFE}_{n,1}$  for a class  $(\mathcal{F}_\lambda^{k,d}, \mathcal{F}_\lambda^{\ell,V,X})$  using an ABIPFE associated with a class  $(\mathcal{F}_\lambda^{k,d}, \mathcal{F}_\lambda^{\ell,V,3X})$ . The ABIPFE should satisfy the structural properties namely *two step decryption* and *linear encryption* as required for the transformation of [2,3]. We describe the properties as follows:

1. *Two step decryption.* An ABIPFE scheme ( $\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec}$ ) admits additional PPT algorithms  $\text{Setup}^*, \text{Dec}_1, \text{Dec}_2$  and an encoding function  $\mathcal{E}$  such that
  - (a) For all  $\lambda, \ell, n, V, X$ ,  $\text{Setup}^*(1^\lambda, \mathcal{F}_\lambda^{\ell,V,X}, \mathcal{F}_\lambda^{k,d}, 1^n)$  uses  $\text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda^{k,d})$  to outputs  $(\text{mpk}, \text{msk})$  where  $\text{mpk}$  includes a bound  $B \in \mathbb{N}$ , a group description  $(\mathbb{G}, \circ)$  of order  $L > n\ell VX$ , which defines an encoding function  $\mathcal{E} : \mathbb{Z}_L \times \mathbb{Z} \rightarrow \mathbb{G}$ .
  - (b) For all  $\mathbf{x} \in \mathbb{Z}^\ell, \mathbf{a} \in \{0, 1\}^k, \text{ct} \leftarrow \text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x})$  and  $\mathbf{y} \in \mathbb{Z}^\ell, f \in \mathcal{F}_\lambda^{k,d}, \text{sk}_{f,\mathbf{y}} \leftarrow \text{KeyGen}(\text{sk}, f, \mathbf{y})$ , we have
$$\text{Dec}_1(\text{mpk}, \text{sk}_{f,\mathbf{y}}, \text{ct}) = \mathcal{E}(\langle \mathbf{x}, \mathbf{y} \rangle \bmod L, \text{noise})$$
for some  $\text{noise} \in \mathbb{N}$ . Furthermore, for all  $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^\ell$  we have  $\Pr[\text{noise} < B] = 1 - \text{negl}(\lambda)$ . We also require that  $\mathcal{E}(\gamma, 0)$  is efficiently computable for any  $\gamma \in \mathbb{Z}_L$ . Moreover, the encoding is linear, that is for  $\gamma, \gamma' \in \mathbb{Z}_L, \text{noise}, \text{noise}' \in \mathbb{Z}$ , we have
$$\mathcal{E}(\gamma, \text{noise}) \circ \mathcal{E}(\gamma', \text{noise}') = \mathcal{E}(\gamma + \gamma' \bmod L, \text{noise} + \text{noise}')$$
  - (c) For all  $\gamma < n\ell VX$  and  $\text{noise} < nB$ ,  $\text{Dec}_2(\mathcal{E}(\gamma, \text{noise})) = \gamma$ .

2. *Linear encryption.* There exists a deterministic algorithm  $\text{Add}$  such that for all  $\mathbf{a} \in \{0, 1\}^k$ ,  $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}^\ell$ , the distributions  $\text{Add}(\text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x}), \mathbf{x}')$  and  $\text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x} + \mathbf{x}' \bmod L)$  are identically distributed. This property will be used in the security proof.

We present the transformation of  $\text{ABMIPFE}_{n,1}$  from  $\text{ABIPFE} = (\text{Setup}', \text{KeyGen}', \text{Enc}', \text{Dec}')$  which satisfies the above properties.

$\text{Setup}(1^\lambda, 1^\ell, \mathcal{F}_\lambda^{d,k}, \text{Att})$  : It computes  $(\text{mpk}_i, \text{msk}_i) \leftarrow \text{Setup}^*(1^\lambda, \mathcal{F}_\lambda^{\ell,V,3X}, \mathcal{F}_\lambda^{k,d}, 1^n)$  and samples  $\mathbf{u}_i \leftarrow \mathbb{Z}_L^\ell$  for  $i \in [n]$ . Then it outputs  $(\text{mpk} = \{\text{mpk}_i\}_{i \in [n]}, \text{msk} = (\{\text{msk}_i, \mathbf{u}_i\}_{i \in [n]}), \{\text{ek}_i = \mathbf{u}_i\}_{i \in [n]})$ . We take  $\text{Att} = (\mathbf{a}_1, \dots, \mathbf{a}_n) \in \{0, 1\}^{kn}$  as each party has a single attribute.

$\text{KeyGen}(\text{msk}, \mathcal{S}, (f_i, \mathbf{y}_i)_{i \in \mathcal{S}})$  : If  $f_i(\mathbf{a}_i) = 1$  for some  $i \in \mathcal{S}$  then returns  $\perp$ . Otherwise, it computes  $\text{sk}_{f_i, \mathbf{y}_i} \leftarrow \text{KeyGen}'(\text{msk}_i, f_i, \mathbf{y}_i)$  for  $i \in \mathcal{S}$  and outputs  $(\mathcal{S}, \text{sk}_{f, \mathbf{y}} = (\{\text{sk}_{f_i, \mathbf{y}_i}\}_{i \in \mathcal{S}}, z = \sum_{i \in \mathcal{S}} \langle \mathbf{u}_i, \mathbf{y}_i \rangle))$ . We assume that the secret-key includes a description of  $(f_i, \mathbf{y}_i)_{i \in \mathcal{S}}$ .

$\text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i, \mathbf{x}_i)$  : It returns  $\text{ct}_i \leftarrow \text{Enc}'(\text{mpk}_i, \mathbf{a}_i, \mathbf{x}_i + \text{ek}_i \bmod L)$ .

$\text{Dec}(\text{mpk}, \mathcal{S}, \text{sk}_{f, \mathbf{y}}, \{\text{ct}_i\}_{i=1}^n)$  : It parses  $\text{sk}_{f, \mathbf{y}} = (\{\text{sk}_{f_i, \mathbf{y}_i}\}_{i \in \mathcal{S}}, z)$  and computes  $\zeta_i \leftarrow \text{Dec}_1(\text{mpk}_i, \text{sk}_{f_i, \mathbf{y}_i}, \text{ct}_i)$  for  $i \in \mathcal{S}$ . Then it returns  $\text{Dec}_2(\circ_{i \in \mathcal{S}} \zeta_i \circ \mathcal{E}(-z, 0))$ .

**Correctness.** Let us assume that  $f_i(\mathbf{a}_i) = 0$  for all  $i \in \mathcal{S}$ . By the correctness of  $\text{Dec}_1$  and  $\text{Dec}_2$  of  $\text{ABIPFE}$ , we see  $\zeta_i = \mathcal{E}(\langle \mathbf{x}_i + \mathbf{u}_i, \mathbf{y}_i \rangle \bmod L, \text{noise}_i)$  for all  $i \in \mathcal{S}$  where  $|\text{noise}_i| < B$  with high probability. Since  $z = \sum_{i \in \mathcal{S}} \langle \mathbf{u}_i, \mathbf{y}_i \rangle$ , by the linearity of  $\mathcal{E}$ , we have  $\circ_{i \in \mathcal{S}} \zeta_i \circ \mathcal{E}(-z, 0) = \mathcal{E}(\sum_{i \in \mathcal{S}} \langle \mathbf{x}_i + \mathbf{u}_i, \mathbf{y}_i \rangle - z \bmod L, \text{noise}) = \mathcal{E}(\sum_{i \in \mathcal{S}} \langle \mathbf{x}_i, \mathbf{y}_i \rangle \bmod L, \text{noise})$  where  $|\text{noise}| < nB$ . Finally,  $|\sum_{i \in \mathcal{S}} \langle \mathbf{x}_i, \mathbf{y}_i \rangle| < L$  implies  $\text{Dec}_2(\circ_{i \in \mathcal{S}} \zeta_i \circ \mathcal{E}(-z, 0))$  returns  $\sum_{i \in \mathcal{S}} \langle \mathbf{x}_i, \mathbf{y}_i \rangle$ .

**Theorem 7** *Assuming the single input  $\text{ABIPFE}$  is Sel-IND secure (respectively,  $Q$ -bounded coSel-IND secure) for a class  $(\mathcal{F}_\lambda^{k,d}, \mathcal{F}_\lambda^{\ell,V,3X})$ , then the above construction of  $\text{ABMIPFE}_{n,1}$  for the class  $(\mathcal{F}_\lambda^{k,d}, \mathcal{F}_\lambda^{\ell,V,X})$  is Adp-IND secure (respectively,  $Q$ -bounded coAdp-IND secure). More specifically, for any PPT adversary  $\mathcal{A}$ , there exists a PPT adversary  $\mathcal{B}$  such that*

$$\text{Adv}_{\text{ABMIPFE}_{n,1}, \mathcal{A}}^{\text{xx-IND}}(\lambda) \leq n \cdot \text{Adv}_{\text{ABIPFE}, \mathcal{B}}^{\text{yy-IND}}(\lambda) + \text{negl}(\lambda)$$

where  $(\text{xx}, \text{yy}) \in \{(\text{Adp}, \text{Sel}), (\text{coAdp}, \text{coSel})\}$ .

We prove this Theorem in App. A. Our 1-bounded coSel-IND secure  $\text{ABIPFE}$  of Sec. 4 can be fit into the above transformation. Formally, we state the result in the following corollary.

**Corollary 1** *Assuming  $\text{LWE}_{q,\alpha,n}$  is hard with  $q, \alpha, n$  are as defined at the end of Sec. 4, there exists a  $\kappa$ -bounded coAdp-IND secure  $\text{ABMIPFE}_{\kappa,1}$  scheme.*

*Proof.* We instantiate the generic construction of the  $\text{ABMIPFE}_{\kappa,1}$  with our single input  $\text{ABIPFE}$  of Sec. 4. Since the  $\text{ABIPFE}$  is 1-bounded coSel-IND secure, the instantiation yields  $\kappa$ -bounded coAdp-IND secure  $\text{ABMIPFE}_{\kappa,1}$  where the

adversary is allowed to submit a single function for each encryption slot before the setup phase. Let  $f_1, \dots, f_\kappa$  be the functions. Then the oracle  $\mathcal{O}_{KG}(\cdot, \cdot)$  takes input  $(S, \{\mathbf{y}\}_{i \in S})$  and outputs  $(S, \{\text{KeyGen}'(\text{msk}_i, f_i, \mathbf{y}_i)\}_{i \in S}, \sum_{i \in S} \langle \mathbf{u}_i, \mathbf{y}_i \rangle)$ . We show that our ABIPFE satisfies the properties namely two step decryption and linear encryption as needed for the transformation. Then, the Corollary will follow by combining Theorems 6 and 7.

1. *Two step decryption.* For the scheme ABIPFE = (Setup, KeyGen, Enc, Dec), we modify the algorithms as follows:
  - (a) The algorithm  $\text{Setup}^*(1^\lambda, \mathcal{F}_\lambda^{\ell, V, X}, \mathcal{F}_\lambda^{k, d}, 1^\kappa)$  works in the same way as Setup except that it sets  $K = \kappa \ell V X$  and the master public-key mpk includes the bound  $B = \lfloor \frac{q}{K} \rfloor$ ,  $L = q$ , the group  $(\mathbb{G}, \circ) = (\mathbb{Z}, +)$  which defines the encoding function  $\mathcal{E} : \mathbb{Z}_L \times \mathbb{Z} \rightarrow \mathbb{Z}$  as
$$\mathcal{E}(\gamma \bmod q, \text{noise}) = \gamma \cdot \lfloor \frac{q}{K} \rfloor + \text{noise} \bmod q$$
for all  $\gamma \in \mathbb{Z}_q, \text{noise} \in \mathbb{Z}$ .
  - (b) For all  $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^\ell, f \in \mathcal{F}_\lambda^{k, d}, \text{ct} \leftarrow \text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x})$  and  $\text{sk}_{f, \mathbf{y}} \leftarrow \text{KeyGen}(\text{sk}, f, \mathbf{y})$  we have

$$\begin{aligned} \text{Dec}_1(\text{mpk}, \text{sk}_{f, \mathbf{y}}, \text{ct}) &= \mathbf{y}^\top \text{ct}_2 - \text{sk}_{f, \mathbf{y}}^\top \cdot (\mathbf{c}_0 | \mathbf{c}_f) \\ &= \left\lfloor \frac{q}{K} \right\rfloor \cdot \langle \mathbf{x}, \mathbf{y} \rangle + \mathbf{y}^\top (\mathbf{e}_2 + \mathbf{e}_3) - (\mathbf{R}_f \mathbf{y})^\top (\mathbf{e}_1 | \mathbf{e}_f) \\ &= \mathcal{E}(\langle \mathbf{x}, \mathbf{y} \rangle \bmod q, \text{noise}) \end{aligned}$$

where  $\text{ct}_1 = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_k) \in (\mathbb{Z}_q^m)^{k+1}, \text{ct}_2 \in \mathbb{Z}_q^\ell$  and  $\mathbf{c}_f = \text{Eval}_{\text{ct}}(f, ((a_i, \mathbf{B}_i, \mathbf{c}_i)_{i=1}^k))$ . That is,  $\text{Dec}_1$  follows step 1 to 3 as in the Dec algorithm. We note that, by the correctness of ABIPFE we have  $\Pr[\text{noise} < B] = 1 - \text{negl}(\lambda)$ . Furthermore, one can easily verify that  $\mathcal{E}(\gamma, 0)$  is efficiently computable for any  $\gamma \in \mathbb{Z}_q$  and the encoding  $\mathcal{E}$  is linear.

- (c) From the correctness of decryption, for all  $\gamma < \kappa \ell V X$  and  $\text{noise} < \kappa B$  we have  $\text{Dec}_2(\mathcal{E}(\gamma \bmod q, \text{noise})) = \gamma$ .
2. *Linear encryption.* For all  $\mathbf{x}' \in \mathbb{Z}^\ell$  and  $(\text{ct}_1, \text{ct}_2) \in (\mathbb{Z}_q)^{m(k+1)} \times \mathbb{Z}_q^\ell$ , we define  $\text{Add}((\text{ct}_1, \text{ct}_2), \mathbf{x}') = (\text{ct}_1, \text{ct}_2 + \mathbf{x}' \cdot \lfloor \frac{q}{K} \rfloor \bmod q)$ . Then, for all  $\mathbf{a} \in \{0, 1\}^k, \mathbf{x}, \mathbf{x}' \in \mathbb{Z}^\ell$  and  $(\text{ct}_1, \text{ct}_2) = (\mathbf{H}_\mathbf{a}^\top \mathbf{s} + \mathbf{v}, \mathbf{D}^\top \mathbf{s} + \mathbf{e}_2 + \mathbf{e}_3 + \lfloor \frac{q}{K} \rfloor \cdot \mathbf{x}) = \text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x})$  we observe that

$$\begin{aligned} \text{Add}((\text{ct}_1, \text{ct}_2), \mathbf{x}') &= (\text{ct}_1, \mathbf{D}^\top \mathbf{s} + \mathbf{e}_2 + \mathbf{e}_3 + \lfloor \frac{q}{K} \rfloor \cdot (\mathbf{x} + \mathbf{x}') \bmod q) \\ &= \text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x} + \mathbf{x}' \bmod q) \end{aligned}$$

This proves that  $\text{Add}(\text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x}), \mathbf{x}')$  and  $\text{Enc}(\text{mpk}, \mathbf{a}, \mathbf{x} + \mathbf{x}' \bmod q)$  are identically distributed.

## 6 Conclusion

We have shown the way of embedding any polynomial-size boolean circuit into the secret-keys of the existing IPFE scheme [6] and its multi-input variants

[2]. The secret-keys are short and both the secret-keys and ciphertexts of our ABIPFEs depend on the depth of the circuits. Moreover, the security is based on LWE assumption which makes our ABIPFEs post-quantum secure. The notion of 1-bounded **coSel-IND** security permits the adversary to query many secret-keys that can decrypt the challenge ciphertexts. This delivers a partial solution to the open problem in the key-policy setting given by Abdalla et al. [3]. We have seen that 1-bounded **coSel-IND** secure single input ABIPFE is sufficient to build a  $\kappa$ -bounded **coAdp-IND** ( $\kappa > 1$ ) secure attribute-based multi-input IPFE which delivers more finer access control.

However, the secret-keys of our ABIPFE that decrypt challenge messages are all corresponding to a single function. Achieving  $Q$ -bounded **coSel-IND** security with  $Q > 1$  or (stronger) **Sel-IND** security for ABIPFE is a challenging open problem. Other than strengthening the security of ABIPFE, we can also investigate decentralized ABIPFE [35], attribute-based access control in case of unbounded IPFE [21] or traceable ABIPFE [19] for a specific class of policies.

## References

1. M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *IACR International Workshop on Public Key Cryptography*, pages 733–751. Springer, 2015.
2. M. Abdalla, D. Catalano, D. Fiore, R. Gay, and B. Ursu. Multi-input functional encryption for inner products: function-hiding realizations and constructions without pairings. In *Annual International Cryptology Conference*, pages 597–627. Springer, 2018.
3. M. Abdalla, D. Catalano, R. Gay, and B. Ursu. Inner-product functional encryption with fine-grained access control. *IACR Cryptol. ePrint Arch.*, 2020:577, 2020.
4. S. Agrawal, S. Bhattacharjee, D. H. Phan, D. Stehlé, and S. Yamada. Efficient public trace and revoke from standard assumptions. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2277–2293, 2017.
5. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (h) ibe in the standard model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 553–572. Springer, 2010.
6. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *Annual International Cryptology Conference*, pages 333–362. Springer, 2016.
7. M. Ajtai. Generating hard instances of the short basis problem. In *International Colloquium on Automata, Languages, and Programming*, pages 1–9. Springer, 1999.
8. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. 2009.
9. P. Ananth, A. Jain, D. Khurana, and A. Sahai. Indistinguishability obfuscation without multilinear maps: io from lwe, bilinear maps, and weak pseudorandomness. *IACR Cryptol. ePrint Arch.*, 2018:615, 2018.
10. P. Ananth, A. Jain, H. Lin, C. Matt, and A. Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In *Annual International Cryptology Conference*, pages 284–332. Springer, 2019.

11. P. Ananth and A. Sahai. Functional encryption for turing machines. In *Theory of Cryptography Conference*, pages 125–153. Springer, 2016.
12. B. Barak, S. B. Hopkins, A. Jain, P. Kothari, and A. Sahai. Sum-of-squares meets program obfuscation, revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 226–250. Springer, 2019.
13. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.
14. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 533–556. Springer, 2014.
15. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference*, pages 253–273. Springer, 2011.
16. J.-Y. Cai. A relation of primal-dual lattices and the complexity of shortest lattice vector problem. *Theoretical Computer Science*, 207(1):105–116, 1998.
17. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 523–552. Springer, 2010.
18. Y. Chen, L. Zhang, and S.-M. Yiu. Practical attribute based inner product functional encryption from simple assumptions. *IACR Cryptol. ePrint Arch.*, 2019:846, 2019.
19. X. T. Do, D. H. Phan, and D. Pointcheval. Traceable inner product functional encryption. In *Cryptographers Track at the RSA Conference*, pages 564–585. Springer, 2020.
20. L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *Annual Cryptology Conference*, pages 335–352. Springer, 2014.
21. E. Dufour-Sans and D. Pointcheval. Unbounded inner-product functional encryption with succinct keys. In *International Conference on Applied Cryptography and Network Security*, pages 426–441. Springer, 2019.
22. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.
23. S. Garg, E. Miles, P. Mukherjee, A. Sahai, A. Srinivasan, and M. Zhandry. Secure obfuscation in a weak multilinear map model. In *Theory of Cryptography Conference*, pages 241–268. Springer, 2016.
24. R. Gay, A. Jain, H. Lin, and A. Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. *IACR Cryptol. ePrint Arch*, 2020:764, 2020.
25. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008.
26. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.
27. A. Jain, H. Lin, C. Matt, and A. Sahai. How to leverage hardness of constant-degree expanding polynomials over  $\mathbb{R}$  to build io. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 251–281. Springer, 2019.



28. A. Jain, H. Lin, and A. Sahai. Simplifying constructions and assumptions for io. Technical report, Technical report, Cryptology ePrint Archive, Report 2019/1252, 2019. [https](https://eprint.iacr.org/2019/1252) , 2019.
29. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: more compact ibes from ideal lattices and bilinear maps. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 682–712. Springer, 2016.
30. A. Lombardi and V. Vaikuntanathan. Limits on the locality of pseudorandom generators and applications to indistinguishability obfuscation. In *Theory of Cryptography Conference*, pages 119–137. Springer, 2017.
31. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.
32. A. O’Neill. Definitional issues in functional encryption. *IACR Cryptol. ePrint Arch.*, 2010:556, 2010.
33. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93, 2005.
34. A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–473. Springer, 2005.
35. Z. Wang, X. Fan, and F.-H. Liu. Fe for inner products and its application to decentralized abe. In *IACR International Workshop on Public Key Cryptography*, pages 97–127. Springer, 2019.
36. B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *Annual International Cryptology Conference*, pages 619–636. Springer, 2009.

## A Security Proof of Theorem 7

We use the two step technique of [2,3] to prove the theorem. We will prove the adaptive security for  $\text{ABMIPFE}_{n,1}$ , as the co-adaptive part can be proved similarly. Since we are in the public-key setting there is a standard reduction from *one*-challenge to *many*-challenge security. Therefore, any single input (public-key) ABIPFE which is proved secure in one-challenge setting can be used as a many-challenge secure ABIPFE.

First, we show that the advantage of the adversary  $\mathcal{A}$  in *one-time security* with its associated game  $\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{ONE-Adp-IND}}(1^\lambda, b)$  is zero where by one-time security we mean that  $\mathcal{A}$  can make at most one query to  $\mathcal{O}_{\text{Enc}}(\cdot, \cdot, \cdot)$  of the form  $(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$  for each slot  $i \in [n]$ . Let  $\mathbf{G}_0$  be the experiment  $\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{ONE-Adp-IND}}(1^\lambda, b)$  with  $b \leftarrow \{0, 1\}$  except that the challenger guesses all the queries to  $\mathcal{O}_{\text{Enc}}(\cdot, \cdot, \cdot)$  of the form  $(i, \mathbf{w}_i^0, \mathbf{w}_i^1)$  for each  $i \in [n]$  in advance, where  $(\mathbf{w}_i^0, \mathbf{w}_i^1) \leftarrow \{0, \dots, 3X - 1\}^\ell$ . Then, we have  $\text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\mathbf{G}_0}(\lambda) = (3X)^{2n\ell} \cdot \text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{ONE-Adp-IND}}(\lambda)$ .

Let  $(i, \mathbf{x}_i^0, \mathbf{x}_i^1)$  be the actual query made by  $\mathcal{A}$  for each  $i \in [n]$ . In  $\mathbf{G}_0$  we have  $\mathbf{w}_i^0 = \mathbf{x}_i^0$  and  $\mathbf{w}_i^1 = \mathbf{x}_i^1$  for each  $i \in [n]$ . We show that  $\text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\mathbf{G}_0}(\lambda) = 0$  by utilizing the fact that the distributions  $\{\mathbf{u}_i\}_{i \in [n]}$  and  $\{\mathbf{u}_i + \mathbf{w}_i^1 - \mathbf{w}_i^0\}$  are identically distributed for  $\mathbf{u}_i \leftarrow \mathbb{Z}_L^\ell$ . If  $b = 0$ , then the first distribution corresponds to the challenge messages of the form  $\{\text{Enc}'(\text{mpk}_i, \mathbf{a}_i, \mathbf{x}_i^0 + \mathbf{u}_i \bmod L)\}_{i \in [n]}$  and the secret-keys are of the form  $\{(\mathbf{S}, \{\text{KeyGen}'(\text{msk}_i, f_i, \mathbf{y}_i)\}_{i \in [n]}, \sum_{i \in [n]} \langle \mathbf{u}_i, \mathbf{y}_i \rangle)\}$ . When  $b = 1$ , the second distribution corresponds to the challenge messages of the form  $\{\text{Enc}'(\text{mpk}_i, \mathbf{a}_i, \mathbf{x}_i^0 + \mathbf{u}_i + \mathbf{x}_i^1 - \mathbf{x}_i^0 \bmod L) = \text{Enc}'(\text{mpk}_i, \mathbf{a}_i, \mathbf{x}_i^1 + \mathbf{u}_i \bmod L)\}_{i \in [n]}$  and the secret-keys are of the form  $\{(\mathbf{S}, \{\text{KeyGen}'(\text{msk}_i, f_i, \mathbf{y}_i)\}_{i \in [n]}, \sum_{i \in [n]} \langle \mathbf{u}_i + \mathbf{x}_i^1 - \mathbf{x}_i^0, \mathbf{y}_i \rangle = \sum_{i \in [n]} \langle \mathbf{u}_i, \mathbf{y}_i \rangle)\}$ . Note that, the key queries of  $\mathcal{A}$  should satisfy that  $\sum_{i \in [n]} \langle \mathbf{x}_i^0, \mathbf{y}_i \rangle = \sum_{i \in [n]} \langle \mathbf{x}_i^1, \mathbf{y}_i \rangle$ . This proves the statistical indistinguishability between these two distributions. Hence,  $\text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\mathbf{G}_0}(\lambda) = 0$  and hence  $\text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{ONE-Adp-IND}}(\lambda) = 0$ .

We now prove the above theorem using the following two games.

**Game 1.** This game is the same as the standard experiment  $\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 0)$  except that the encryption oracle  $\mathcal{O}_{\text{Enc}}(i, (\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}))$  now returns  $\text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i, \mathbf{x}_i^{j,0} + \mathbf{x}_i^{1,0} - \mathbf{x}_i^{1,1})$  for all  $i \in [n]$ . We show that there exists a PPT adversary  $\mathcal{B}_1$  such that the advantage of  $\mathcal{A}$  in distinguishing between  $\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 0)$  and game 1 is upper bounded by  $\text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{B}_1}^{\text{ONE-Adp-IND}}(\lambda)$ . Let  $\text{mpk}$  be the master public-key that  $\mathcal{B}_1$  receives from its challenger. Then  $\mathcal{B}_1$  simulates the game for  $\mathcal{A}$  as follows.

$\mathcal{B}_1(1^\lambda, \text{mpk})$ :

1.  $\mathcal{B}_1$  sends  $\text{mpk}$  to  $\mathcal{A}$ .
2. Whenever  $\mathcal{A}$  queries to  $\mathcal{O}_{\text{corr}}$  and  $\mathcal{O}_{\text{KG}}$ ,  $\mathcal{B}_1$  uses its own oracles to answer.
3. For each slot  $i \in [n]$ , let  $(i, (\mathbf{x}_i^{1,0}, \mathbf{x}_i^{1,1}))$  be the first query to  $\mathcal{O}_{\text{Enc}}$ . Then  $\mathcal{B}_1$  forwards it to its own encryption oracle and receives a ciphertext  $\text{ct}_i^* \leftarrow \text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i, \mathbf{x}_i^{1,b})$  for some  $b \leftarrow \{0, 1\}$ . From the next query  $(i, (\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}))$  with  $j > 1$  to the  $i$ -th slot  $\mathcal{B}_1$  sends  $\text{Add}(\text{ct}_i^*, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0})$  to  $\mathcal{A}$ .
4. Finally,  $\mathcal{B}_1$  returns the bit  $b$  which it receives from  $\mathcal{A}$ .

Observe that,  $\text{Add}(\text{ct}_i^*, \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0}) = \text{Enc}'(\text{mpk}_i, \mathbf{a}_i, \mathbf{x}_i^{1,b} + \mathbf{u}_i + \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} \bmod L) = \text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{x}_i^{1,b} + \mathbf{x}_i^{j,0} - \mathbf{x}_i^{1,0} \bmod L)$ . Therefore, when  $b = 0$ , the adversary  $\mathcal{B}_1$  simulates  $\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 0)$  and if  $b = 1$ , then  $\mathcal{B}_1$  simulates game 1. This proves that

$$\left| \Pr[\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 0) = 1] - \Pr[\text{Expt}_{\mathcal{A}}^{\text{Game 1}}(1^\lambda) = 1] \right| < \text{Adv}_{\text{ABMIPFE}_{n,1},\mathcal{B}_1}^{\text{ONE-Adp-IND}}(\lambda)$$

**Game 2.** This is the standard experiment  $\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 1)$ . We show that there exists a PPT adversary  $\mathcal{B}$  such that

$$\left| \Pr[\text{Expt}_{\mathcal{A}}^{\text{Game 1}}(1^\lambda) = 1] - \Pr[\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 1) = 1] \right| < n \cdot \text{Adv}_{\text{ABIPFE},\mathcal{B}}^{\text{Sel-IND}}(\lambda) + \text{negl}(\lambda)$$

We switch the distribution of the ciphertext one slot at a time depending on the security of ABIPFE scheme. Let  $v \in [n]$  and  $\mathcal{B}_v$  be the adversary against the ABIPFE employed for the  $v$ -th slot. After receiving the master public-key  $\text{mpk}_v$ ,  $\mathcal{B}_v$  simulates the game for  $\mathcal{A}$  as follows.

$\mathcal{B}_v(1^\lambda, 1^\ell, \mathcal{F}_\lambda^{d,k}, \text{Att}, \text{mpk}_v)$ :

1. It computes  $(\text{mpk}_i, \text{msk}_i) \leftarrow \text{Setup}^*(1^\lambda, \mathcal{F}_\lambda^{\ell,V,X}, \mathcal{F}_\lambda^{k,d}, 1^n)$  for  $i \in [n] \setminus \{v\}$  and samples  $\mathbf{u}_i \leftarrow \mathbb{Z}_L^\ell$  for  $i \in [n]$ . Then it sets  $\{\text{ek}_i = \mathbf{u}_i\}_{i \in [n]}$  and sends  $\text{mpk} = \{\text{mpk}_i\}_{i \in [n]}$  to  $\mathcal{A}$ .
2.  $\mathcal{B}_v$  uses  $\{\mathbf{u}_i\}_{i \in [n]}$  to answer all queries to  $\mathcal{O}_{\text{Corr}}(\cdot)$ .
3. When  $\mathcal{B}_v$  receives a secret-key query for  $(S, (f_i, \mathbf{y}_i)_{i \in S})$ , it first sends  $(f_v, \mathbf{y}_v)$  to its challenger if  $v \in S$  and gets a secret-key  $\text{sk}_{f_v, \mathbf{y}_v}$ . For all  $i \in S \setminus \{v\}$ ,  $\mathcal{B}_v$  computes  $\text{sk}_{f_i, \mathbf{y}_i} \leftarrow \text{KeyGen}'(\text{msk}_i, f_i, \mathbf{y}_i)$  and sends  $(\text{sk}_{f_i, \mathbf{y}_i})_{i \in S}, z = \sum_{i \in S} \langle \mathbf{u}_i, \mathbf{y}_i \rangle$  to  $\mathcal{A}$ .
4. For each query  $(i, (\mathbf{x}_i^{j,0}, \mathbf{x}_i^{j,1}))$  to  $\mathcal{O}_{\text{Enc}}(\cdot, \cdot, \cdot)$ ,  $\mathcal{B}_v$  generates the challenge ciphertext as follows:
  - (a) If  $i < v$ ,  $\mathcal{B}_v$  sends  $\text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i, \mathbf{x}_i^{j,1})$ .
  - (b) If  $i > v$ ,  $\mathcal{B}_v$  sends  $\text{Enc}(\text{mpk}, \text{ek}_i, \mathbf{a}_i, \mathbf{x}_i^{j,0} + \mathbf{x}_i^{1,1} - \mathbf{x}_i^{1,0})$ .
  - (c) If  $i = v$ ,  $\mathcal{B}_v$  sends the challenge messages  $(\mathbf{x}_v^{j,0} + \mathbf{x}_v^{1,1} - \mathbf{x}_v^{1,0} \bmod L, \mathbf{x}_v^{j,1} \bmod L)$  to its own challenger and gets back a ciphertext  $\text{ct}_v^*$ . Finally, it sends  $\text{Add}(\text{ct}_v^*, \mathbf{u}_v)$  to  $\mathcal{A}$ .

To rely on the Sel-IND security of ABIPFE we need to show  $\langle \mathbf{x}_v^{j,0} + \mathbf{x}_v^{1,1} - \mathbf{x}_v^{1,0}, \mathbf{y}_v \rangle = \langle \mathbf{x}_v^{j,1}, \mathbf{y}_v \rangle \bmod L$  or  $\langle \mathbf{x}_v^{j,0} - \mathbf{x}_v^{1,0}, \mathbf{y}_v \rangle = \langle \mathbf{x}_v^{j,1} - \mathbf{x}_v^{1,1}, \mathbf{y}_v \rangle \bmod L$  for all  $j$  queried by  $\mathcal{A}$ . In fact, this holds due to the restriction on the queries made by  $\mathcal{A}$  as given in the security definition 4 of ABMIPFE.

Therefore, the advantage of  $\mathcal{A}$  in distinguishing between the intermediate games is upper bounded by the advantage of  $\mathcal{B}_v$  in the Sel-IND security experiment of ABIPFE scheme. By combining all the  $n$  intermediate advantages we get a PPT adversary  $\mathcal{B}$  such that

$$\left| \Pr[\text{Expt}_{\mathcal{A}}^{\text{Game 1}}(1^\lambda) = 1] - \Pr[\text{Expt}_{\text{ABMIPFE}_{n,1},\mathcal{A}}^{\text{Adp-IND}}(1^\lambda, 1) = 1] \right| < n \cdot \text{Adv}_{\text{ABIPFE},\mathcal{B}}^{\text{Sel-IND}}(\lambda) + \text{negl}(\lambda)$$

This proves the theorem.