Lattice-based IBE with Equality Test Supporting Flexible Authorization in the Standard Model

Giang Linh Duc Nguyen[3]([✉]), Willy Susilo[1]([✉]), Dung Hoang Duong[1]([✉]), Huy Quoc Le[1,2]([✉]) and Fuchun Guo[1]([✉])

[1] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong
Northfields Avenue, Wollongong NSW 2522, Australia
{hduong,wsusilo,fuchun}@uow.edu.au, qhl576@uowmail.edu.au
[2] CSIRO Data61, Sydney, NSW, Australia
[3] Futurify Software Development
17 Street No. 2, Ward 4, District 3, Ho Chi Minh City, Vietnam
ndlgiang.edu@gmail.com

Abstract. Identity-based encryption with equality test supporting flexible authorization (IBEET-FA) allows the equality test of underlying messages of two ciphertexts while strengthens privacy protection by allowing users (identities) to control the comparison of their ciphertexts with others. IBEET by itself has a wide range of useful applicable domain such as keyword search on encrypted data, database partitioning for efficient encrypted data management, personal health record systems, and spam filtering in encrypted email systems. The flexible authorization will enhance privacy protection of IBEET. In this paper, we propose an efficient construction of IBEET-FA system based on the hardness of learning with error (LWE) problem. Our security proof holds in the standard model.

Key words: identity-based encryption, equality test, flexible authorization, lattice-based cryptography, Learning with Errors.

1 Introduction

The concept of identity-based encryption with equality test supporting flexible authorization (IBEET-FA) is the combination of public key encryption with equality test supporting flexible authorization (PKEET-FA) and identity-based encryption (IBE). IBE is a public key encryption mechanism where a public key can be an arbitrary string such as email address or phone number. The administrator who has knowledge of master secret key then can generate the corresponding private key by using Private-Key Generator (PKG). IBE by itself can simplify the certificate management of public key encryption. Identity-based encryption scheme with equality test (IBEET) was first introduced by Ma [12]. It supports the equality test of underlying message of two ciphertext with the help of trapdoors. Hence IBEET has found a wide range of applications especially in

smart city applications [18], in cloud computing such as partition of encrypted emails [20], malware detection and verifiability of encrypted data [21], and in wireless body area networks [22]. The flexible authorization (FA) notion was first introduced by Ma et al. [13]. It enhances the privacy protection of data owners by allowing the users (identities) more choices in controlling of what they want to compare, either between identities' ciphertexts, between two ciphertexts or between a specific ciphertext and other identity's ciphertexts.

One particular application scenario of using IBEET in smart city is in smart healthcare. Assume that, each family in the smart city has a smart device for monitoring their daily health, such as sleeping time, heart beat rate, how many steps a person walk a day, and so on. Their data is encrypted and stored in the cloud sever managed by the government and the encrypted data can be accessed by the healthcare department from the government. It is a common request that the healthcare officers need to get some residents' health information in order to monitor the community's health. The officers then can perform equality test for specific keywords on encrypted data stored in the cloud server. For this scenario, we can see that Type-1 is needed since this type of authorization allows the officers to do equality test on all ciphertexts from the users in the community.

Let consider the case in which some third party, such as a medicine company, can harm the privacy if the system (comprising of smart healthcare devices) is only equipped with the Type-1 authorization. Suppose that, the medicine company wishes to know the body mass index (BMI) of people in the community for their new kind of weight loss pill. Obviously, the advertising company can encrypt a specific keyword relating to the BMI (e.g., the key word could be " obese") using any or all public key(s) of users in the community. Therefore, if the advertising companies are given the Type-1 method, the company can perform equality test on all available ciphertexts in the database, which is not good for the residents' privacy. However, it is a standard privacy policy by law that users can opt-in or opt-out for using of their data. Then, in this case, the users can choose the Type-2 and/or Type-3 authorization which definitely help to enhance the privacy protection.

Note that in the identity-based encryption (IBE), we have a reduction from the IND security to the OW security (see [23, Table 1]). However, note that in the case of IBEET (hence, IBEET-FA), we do not always have the reduction. The reason for this is that, an OW adversary is able to get the trapdoor according to the target user, while an IND adversary is not allowed to get the target user's trapdoor. Then both IND and OW should be simultaneously required for the IBEET-FA. Moreover, choosing CPA or CCA, CCA2 security depends on the real-life applications to which the IBEET-FA is applied. Recall that, the CPA security ensures that, even if the attacker can get ciphertexts for arbitrary plaintexts that it wants, the attacker's advantage in revealing all or part of the secret encryption key is negligible. While the CCA and CCA2 securities guarantee that, even if the attackers can obtain the ciphertexts and the corresponding plaintexts, their advantage in attacks is still negligible. For IBEET and IBEET-FA, it is folkore to consider the security model which involves IND-

ID-CPA (indistinguishability under chosen-identity and under chosen-plaintext attacks) and OW-ID-CPA (one-wayness under chosen-identity and under chosen-plaintext attacks) into account.

Related Works. There have been a lot of works involving with PKEET and IBEET such as Lee et al. [9], Wu et al. [18] and Duong et al. [7] just to name a few. Lee et al. [9] inherited knowledge from the existing proposals proven secure under the hardness of number theoretic assumptions and in the random oracle (ROM). The authors introduced a generic construction for PKEET (and also for IBEET) which is secure in the standard model (SDM) being able to be instantiated in the lattice setting. Their constructions based on two-level hierarchical identity-based, a strong unforgeable one time signature and a cryptographic hash function. Wu et al. [18] proposed an efficient IBEET scheme secure in the random oracle model (ROM) replying on the Computational Diffie-Hellman (CDH) assumption. Duong et al. [7] has introduced an efficient direct construction of an IBEET based on lattices which is secure in the standard model (SDM). Their method exploits the adaptive identity-based encryption (Full-IBE) scheme which is a post-quantum instantiation based on lattices proposed by [1].

Regarding the encryptions that support equality test with FA, there have been a few of works in the literature, e.g., Ma et al. [13], Duong et al. [8]. As the first PKEET-FA, Ma et al. [13] enhanced the original PKEET scheme which lacks of a solution for users to control the comparison of their ciphertexts with others' by adding authorization mechanism. The authors also introduced new definitions, security policies, and a PKEET construction supporting four type of authorization proven under the CDH assumption and in the ROM. Recently, Duong et al. [8] have studied Ma et al. works and proposed the first quantum-safe PKEET supporting flexible authorization which are secure in the SDM based on the integer lattice and ideal lattice hard problems.

Table 1: A summary of some PKEET and IBEET with/without flexible authorization. Here DBDH stands for the decisional Billinear Diffie-Hellman assumption.

Literature	PKEET/IBEET	Hard Assumption	Security Model	with FA?
Ma et al. [13]	PKEET	CDH	ROM	✓
Lee et al. [9]	PKEET & IBEET	CDH+DBDH or LWE	SDM	✗
Wu et al. [18]	IBEET	CDH	ROM	✗
Duong et al. [7]	IBEET	LWE	SDM	✗
Duong et al. [8]	PKEET	LWE	SDM	✓
This work	IBEET	LWE	SDM	✓

We summarize some related works in Table 1 below. According to the best of our knowledge, there have not been any post-quantum secure IBEET that sup-

Table 2: Comparison of our IBEET-FA with other IBEET constructions. Data sizes are in number of field elements. Here t is the length of messages, ℓ is the length of identity.

Scheme	Ciphertext	Public key	Master SK	Secret key
Our proposed IBEET-FA	$m^2 + 2t + 6m + \lambda$	$(\ell+3)mn + nt$	$2m^2$	$4m^2$
IBEET of Duong et al. [7]	$2t + 4m$	$(\ell+3)mn + nt$	$2m^2$	$4mt$
IBEET of Lee et al. [9]	$8m + 2t + 2mt$	$(\ell+3)mn + nt$	$2m^2$	$2mt$

ports a flexible authorization mechanism. Therefore, it is necessary to construct an identity-based encryption with equality test supporting flexible authorization in the standard model and secure in quantum era.

Our contribution. We propose the first concrete construction of an IBEET-FA scheme secure in the standard model based on the hardness assumption of lattices. In particular, our IBEET-FA construction offers the IND-ID-CPA and OW-ID-CPA security. However, we emphasize that our scheme can achieve CCA2 using the BCHK's transformation [4] with the cost of increasing ciphertext size.

Technical Overview. Our idea is from the works of Duong et al. [7, 8]. Recall that, the PKEET-FA of Duong et al. [8] exploits the full-IBE construction by Agrawal et al. [1]. In particular:

- Each user has their own pair of (PK, SK).
- In Encrypt method, they choose a vector tag $\mathbf{b} = (b_1, \cdots, b_\ell) \in \{-1, 1\}^\ell$ and then compute $F_1 := (A|B + \sum_{i=1}^{\ell} b_i A_i)$. That means each user can produce many ciphertexts, each of which contains one different vector tag $\mathbf{b}$.
- The vector tag $\mathbf{b}$ is then combined with the trapdoor (which is the second part of the secret key SK) for flexible authorization methods.

Although our IBEET-FA work is inspired from [8] and [1], we face the following obstacles:

- The whole system only has one pair of (MPK, MSK).
- Each user has their own ID $= (\mathsf{id}_1, \cdots, \mathsf{id}_\ell) \in \{-1, 1\}^\ell$, which is used to compute the matrix $F_{\mathsf{ID}} = (A|B + \sum_{i=1}^{\ell} \mathsf{id}_i A_i)$ in Extract algorithm to produce their secret key from MSK. Each user can produce multiple ciphertexts. However, those all ciphertexts contain the same ID of the user. That's why the ID cannot be used as the role of vector tag $\mathbf{b}$.
- We need a new tag (which can be a random vector or matrix) to play the role of the vector tag $\mathbf{b}$. This tag will be appended to F_{ID} and be stored in ciphertext for flexible authorization later.

Therefore, in order to support FA with many different types, we need to provide additional tag which is random and specified for each ciphertext. Namely, we sample uniformly at random a matrix R (called *matrix tag*) and build the matrix $F = (A|B + \sum_{i=1}^{\ell} \mathsf{id}_i A_i | AR)$ which will be used during the encryption algorithm.

By using the matrix AR like this, we can also construct a reduction from the learning with errors (LWE) problem to the IND-ID-CPA (and OW-ID-CPA) security of the proposed IBEET-FA. If we replace AR by some random matrix, the reduction will not work. In summary, we use the matrix AR for two purposes:

- **P1:** To combine with trapdoor for different Type-α flexible authorisation.
- **P2:** For the reduction from LWE to the IND-ID-CPA and to OW-ID-CPA security to work.

The ciphertext in our scheme has this form $\mathsf{CT} = (R, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5)$, where R is the matrix tag to support flexible authorization, $(\mathbf{c}_1, \mathbf{c}_3)$ is the encryption of message $\mathbf{m}$ and $(\mathbf{c}_2, \mathbf{c}_4)$ is the encryption of hash value $H(\mathbf{m})$. Both R and $(\mathbf{c}_2, \mathbf{c}_4)$ are used to support flexible authorization. Here $\mathbf{c}_5 = H'(\mathbf{c}_1 || \mathbf{c}_2 || \mathbf{c}_3 || \mathbf{c}_4)$ is used to strengthen the FA procedures by checking the consistency of ciphertexts before executing equality test.

We present in Table 2 a comparison of our IBEET-FA with the lattice-based IBEETs of Lee et al. [9] and Duong et al. [7].

The weakness of the existing security model for IBEET and IBEET-FA. We remark that the component $\mathbf{c}_5$ in the ciphertext of our IBEET-FA scheme is only used to check the consistency of ciphertexts before executing equality test. Without $\mathbf{c}_5$ (i.e., assume that $\mathsf{CT} = (R, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$), our scheme still enjoys the IND-ID-CPA and OW-ID-CPA security. However, for real-life applications, our construction with $\mathsf{CT} = (R, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$ may suffer from the following attack. Suppose that we use equality test on encrypted data for a research of counting the number of the obese people, each ciphertext corresponds to each person. Then, if an adversary can create malformed ciphertexts from some original ciphertext by keeping $(R, \mathbf{c}_2, \mathbf{c}_4)$ but replacing the pair of $(\mathbf{c}_1, \mathbf{c}_3)$, then the database would be changed in the number of ciphertexts having the same underlying message, which results in the wrong output of the research. Obviously, this also means that the existing security model for IBEET and IBEET-FA is not strong enough in some manners.

2   Preliminaries

2.1   Identity-based encryption with equality test supporting flexible authorization (IBEET-FA)

In this section, we will recall the model of IBEET-FA and its security model.

Definition 1 *An IBEET-FA scheme consists of the following polynomial-time algorithms:*

- Setup(λ): *On input a security parameter λ, it outputs a public parameter* PP *and a master secret key* MSK.
- Extract(PP, MSK, ID): *On input* PP, MSK *and an identity* ID, *it outputs an identity* ID*'s secret key* SK$_{\mathsf{ID}}$.

- $\mathsf{Enc}(\mathsf{PP}, \mathsf{ID}, \mathbf{m})$: *On input* PP, *an identity* ID *and a message* $\mathbf{m}$, *it outputs a ciphertext* CT.
- $\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{CT})$: *On input* PP, *an identity* ID*'s secret key* $\mathsf{SK}_{\mathsf{ID}}$ *and a ciphertext* CT, *it outputs a message* $\mathbf{m}'$ *or* $\perp$.
  Note that PP *contain the information of the message space* $\mathcal{M}$. *It is used in other algorithms implicitly.*

Suppose that an identity ID_i has secret key SK_i, whose ciphertext is CT_i, and an identity ID_j has secret key SK_j, whose ciphertext is CT_j respectively. The Type-α ($\alpha = 1, 2, 3$) authorization for ID_i and ID_j consist of two algorithms:

- Td_α ($\alpha = 1, 2, 3$): an algorithm to generate trapdoors for ID_i's ciphertexts that are required to be compared with ID_j's ciphertexts.
- Test_α ($\alpha = 1, 2, 3$): an algorithm to check and compare whether two identities' ciphertexts contain the same underlying message or not.

Type-1 Authorization:

- $\mathsf{Td}_1(\mathsf{SK}_{\mathsf{ID}})$: *On input the secret key* $\mathsf{SK}_{\mathsf{ID}}$ *of the identity* ID, *it outputs a trapdoor* $\mathsf{td}_{1,\mathsf{ID}}$ *for that identity.*
- $\mathsf{Test}_1(\mathsf{td}_{1,\mathsf{ID}_i}, \mathsf{td}_{1,\mathsf{ID}_j}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j})$: *On input trapdoors* $\mathsf{td}_{1,\mathsf{ID}_i}, \mathsf{td}_{1,\mathsf{ID}_j}$ *and ciphertexts* $\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j}$ *for identities* $\mathsf{ID}_i, \mathsf{ID}_j$ *respectively, it outputs* 1 *or* 0.

Type-2 Authorization:

- $\mathsf{Td}_2(\mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}_{\mathsf{ID}})$: *On input the secret key* $\mathsf{SK}_{\mathsf{ID}}$ *and ciphertext* $\mathsf{CT}_{\mathsf{ID}}$ *of the identity* ID, *it outputs a trapdoor* $\mathsf{td}_{2,\mathsf{ID}}$ *for* $\mathsf{CT}_{\mathsf{ID}}$ *of that identity.*
- $\mathsf{Test}_2(\mathsf{td}_{2,\mathsf{ID}_i}, \mathsf{td}_{2,\mathsf{ID}_j}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j})$: *On input trapdoors* $\mathsf{td}_{2,\mathsf{ID}_i}, \mathsf{td}_{2,\mathsf{ID}_j}$ *and ciphertexts* $\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j}$ *for identities* $\mathsf{ID}_i, \mathsf{ID}_j$ *respectively, it outputs* 1 *or* 0.

Type-3 Authorization:

- $\mathsf{Td}_3(\mathsf{SK}_{\mathsf{ID}})$: *On input the secret key* $\mathsf{SK}_{\mathsf{ID}}$ *of the identity* ID, *it outputs a trapdoor* $\mathsf{td}_{3,\mathsf{ID}}$ *for identity* ID.
- $\mathsf{Td}_3(\mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}_{\mathsf{ID}})$: *On input the secret key* $\mathsf{SK}_{\mathsf{ID}}$ *and ciphertext* $\mathsf{CT}_{\mathsf{ID}}$ *of the identity* ID, *it outputs a trapdoor* $\mathsf{td}_{3,\mathsf{ID}}$ *for* $\mathsf{CT}_{\mathsf{ID}}$ *of identity* ID.
- $\mathsf{Test}_3(\mathsf{td}_{3,\mathsf{ID}_i}, \mathsf{td}_{3,\mathsf{ID}_j}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j})$: *On input trapdoors* $\mathsf{td}_{3,\mathsf{ID}_i}, \mathsf{td}_{3,\mathsf{ID}_j}$ *and ciphertexts* $\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j}$ *for identities* $\mathsf{ID}_i, \mathsf{ID}_j$ *respectively, it outputs* 1 *or* 0.

Correctness. We say that an IBEET-FA scheme is *correct* if the following conditions hold:

(1) For any security parameter λ, any identity ID and any message $\mathbf{m}$:

$$\Pr\left[\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}_{\mathsf{ID}}) = \mathbf{m} \,\middle|\, \begin{array}{c} \mathsf{SK}_{\mathsf{ID}} \leftarrow \mathsf{Extract}(\mathsf{PP}, \mathsf{MSK}, \mathsf{ID}) \\ \mathsf{CT}_{\mathsf{ID}} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ID}, \mathbf{m}) \end{array}\right] = 1.$$

(2) For any security parameter λ, any identities ID_i, ID_j and any messages $\mathbf{m}_i, \mathbf{m}_j$, it holds that:

$$
\Pr \left[\mathsf{Test} \begin{pmatrix} \mathsf{td}_{\alpha,\mathsf{ID}_i} \\ \mathsf{td}_{\alpha,\mathsf{ID}_j} \\ \mathsf{CT}_{\mathsf{ID}_i} \\ \mathsf{CT}_{\mathsf{ID}_j} \end{pmatrix} = 1 \left| \begin{array}{l} \mathsf{SK}_{\mathsf{ID}_i} \leftarrow \mathsf{Extract}(\mathsf{PP},\mathsf{MSK},\mathsf{ID}_i) \\ \mathsf{CT}_{\mathsf{ID}_i} \leftarrow \mathsf{Enc}(\mathsf{PP},\mathsf{ID}_i,\mathbf{m}_i) \\ \mathsf{td}_{\alpha,\mathsf{ID}_i} \leftarrow \mathsf{Td}_\alpha() \\ \mathsf{SK}_{\mathsf{ID}_j} \leftarrow \mathsf{Extract}(\mathsf{PP},\mathsf{MSK},\mathsf{ID}_j) \\ \mathsf{CT}_{\mathsf{ID}_j} \leftarrow \mathsf{Enc}(\mathsf{PP},\mathsf{ID}_j,\mathbf{m}_j) \\ \mathsf{td}_{\alpha,\mathsf{ID}_j} \leftarrow \mathsf{Td}_\alpha() \end{array} \right. \right] = 1.
$$

if $\mathbf{m}_i = \mathbf{m}_j$, regardless i and j value, where $\alpha = 1,2,3$.

(3) For any security parameter λ, any identities ID_i, ID_j and any messages $\mathbf{m}_i, \mathbf{m}_j$, it holds that:

$$
\Pr \left[\mathsf{Test} \begin{pmatrix} \mathsf{td}_{\alpha,\mathsf{ID}_i} \\ \mathsf{td}_{\alpha,\mathsf{ID}_j} \\ \mathsf{CT}_{\mathsf{ID}_i} \\ \mathsf{CT}_{\mathsf{ID}_j} \end{pmatrix} = 1 \left| \begin{array}{l} \mathsf{SK}_{\mathsf{ID}_i} \leftarrow \mathsf{Extract}(\mathsf{PP},\mathsf{MSK},\mathsf{ID}_i) \\ \mathsf{CT}_{\mathsf{ID}_i} \leftarrow \mathsf{Enc}(\mathsf{PP},\mathsf{ID}_i,\mathbf{m}_i) \\ \mathsf{td}_{\alpha,\mathsf{ID}_i} \leftarrow \mathsf{Td}_\alpha() \\ \mathsf{SK}_{\mathsf{ID}_j} \leftarrow \mathsf{Extract}(\mathsf{PP},\mathsf{MSK},\mathsf{ID}_j) \\ \mathsf{CT}_{\mathsf{ID}_j} \leftarrow \mathsf{Enc}(\mathsf{PP},\mathsf{ID}_j,\mathbf{m}_j) \\ \mathsf{td}_{\alpha,\mathsf{ID}_j} \leftarrow \mathsf{Td}_\alpha() \end{array} \right. \right] = negligible(\lambda)
$$

if for any ciphertexts $\mathsf{CT}_{\mathsf{ID}_i}$, $\mathsf{CT}_{\mathsf{ID}_j}$ and $\mathsf{Dec}(\mathsf{PP},\mathsf{SK}_{\mathsf{ID}_i},\mathsf{CT}_{\mathsf{ID}_i}) \neq \mathsf{Dec}(\mathsf{PP},\mathsf{SK}_{\mathsf{ID}_j},\mathsf{CT}_{\mathsf{ID}_j})$, regardless of whether $i = j$, where $\alpha = 1,2,3$.

Security model of IBEET-FA. For the IBEET-FA security model, we consider two types of adversaries:

- Type-I adversary: the adversaries can perform authorization (equality) tests on the challenge ciphertext by requesting to obtain a trapdoor for authorization of the target identity. Their goal is to reveal the plain text message in the challenge ciphertext.
- Type-II adversary: the adversaries cannot perform authorization (equality) tests on the challenge ciphertext since they cannot obtain a trapdoor for authorization of the target identity. Their goal is to distinguish which message is in the challenge ciphertext between two candidates.

Note that the Type-3 authorization is actually a combination of Type-1 and Type-2 authorizations. Hence the Type-3 authorization queries are intentionally omitted. For simplicity, we only provide Type-α ($\alpha = 1,2$) authorization queries to the adversary in security games. The security model of a IBEET-FA scheme against two types of adversaries above is described as follows.

OW-ID-CPA security against Type-I adversaries. We present the game between a challenger $\mathcal{C}$ and a Type-I adversary $\mathcal{A}$. Let say $\mathcal{A}$ wants to attack target identity ID^* and he can have a trapdoor for all ciphertexts of that identity.

1. **Setup:** $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate the pair $(\mathsf{PP},\mathsf{MSK})$, and sends the public parameter PP to $\mathcal{A}$.

2. **Phase 1:** $\mathcal{A}$ may adaptively make queries many times and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{Ext}}$: on input identity ID ($\neq \mathsf{ID}^*$), $\mathcal{O}^{\mathsf{Ext}}$ returns the ID's secret key $\mathsf{SK}_{\mathsf{ID}}$.
   - $\mathcal{O}^{\mathsf{Td}_\alpha}$ for $\alpha = 1, 2$:
     - oracle $\mathcal{O}^{\mathsf{Td}_1}$ return $\mathsf{td}_{1,\mathsf{ID}} \leftarrow \mathsf{Td}_1(\mathsf{SK}_{\mathsf{ID}})$ using the secret key $\mathsf{SK}_{\mathsf{ID}}$.
     - oracle $\mathcal{O}^{\mathsf{Td}_2}$ return $\mathsf{td}_{2,\mathsf{ID}} \leftarrow \mathsf{Td}_2(\mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}_{\mathsf{ID}})$ using the secret key $\mathsf{SK}_{\mathsf{ID}}$ and $\mathsf{CT}_{\mathsf{ID}}$.
3. **Challenge:** $\mathcal{C}$ chooses a random message $\mathbf{m}$ in the message space and sends $\mathsf{CT}^*_{\mathsf{ID}^*} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ID}^*, \mathbf{m})$ to $\mathcal{A}$.
4. **Phase 2:** same as in Phase 1 except that $\mathcal{A}$ cannot be queried to the secret key extraction oracle $\mathcal{O}^{\mathsf{Ext}}$ for the identity ID^*.
5. **Guess:** $\mathcal{A}$ output $\mathbf{m}'$.

The adversary $\mathcal{A}$ wins the above game if $\mathbf{m} = \mathbf{m}'$ and the success probability of $\mathcal{A}$ is defined as

$$\mathsf{Adv}_{\mathcal{A}, \mathrm{IBEET\text{-}FA}}^{\mathrm{OW\text{-}ID\text{-}CPA}, Type-\alpha}(\lambda) := \Pr[\mathbf{m} = \mathbf{m}'].$$

Remark 2 *It is necessary to have the size of the message space $\mathcal{M}$ to be exponential in the security parameter and the min-entropy of the message distribution is sufficiently higher than the security parameter. If not, a Type-I adversary $\mathcal{A}$ can perform the equality tests with the challenge ciphertext and all other ciphertexts from messages space generated the adversary himself by using the trapdoor for the challenge ciphertext. Then he can reveal the underlying message of the challenge ciphertext in polynomial-time or sufficiently small exponential time in the security parameter. For instance, an algorithm is said to be exponential time, if $T(n)$ is upper bounded by $2^{poly(n)}$, where $poly(n)$ is some polynomial in n. More informally, an algorithm is exponential time if $T(n)$ is bounded by $O(2^{n^k})$. If both n and k are small, then the total time is still sufficiently acceptable.*

IND-ID-CPA security against Type-II adversaries. We illustrate the game between a challenger $\mathcal{C}$ and a Type-II adversary $\mathcal{A}$, who cannot have a trapdoor for all ciphertexts of target identity ID^*.

1. **Setup:** $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate $(\mathsf{PP}, \mathsf{MSK})$ and gives the public parameter PP to $\mathcal{A}$.
2. **Phase 1:** $\mathcal{A}$ may adaptively make queries many times and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{Ext}}$: on input identity ID ($\neq \mathsf{ID}^*$), returns the ID's secret key $\mathsf{SK}_{\mathsf{ID}}$.
   - $\mathcal{O}^{\mathsf{Td}_\alpha}$ for $\alpha = 1, 2$ (where input $\mathsf{ID} \neq \mathsf{ID}^*$):
     - oracle $\mathcal{O}^{\mathsf{Td}_1}$: on input ID, $\mathcal{O}^{\mathsf{Td}_1}$ returns $\mathsf{td}_{1,\mathsf{ID}} \leftarrow \mathsf{Td}_1(\mathsf{SK}_{\mathsf{ID}})$.
     - oracle $\mathcal{O}^{\mathsf{Td}_2}$: on input $(\mathsf{ID}, \mathsf{CT}_{\mathsf{ID}})$, $\mathcal{O}^{\mathsf{Td}_2}$ returns $\mathsf{td}_{2,\mathsf{ID}} \leftarrow \mathsf{Td}_2(\mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}_{\mathsf{ID}})$.
3. **Challenge:** $\mathcal{A}$ selects a target user ID^*, which was never queried to the $\mathcal{O}^{\mathsf{Ext}}$ and $\mathcal{O}^{\mathsf{Td}}$ oracles in Phase 1, and two messages $\mathbf{m}_0$ $\mathbf{m}_1$ of same length and pass to $\mathcal{C}$, who then selects a random bit $b \in \{0, 1\}$, runs $\mathsf{CT}^*_{\mathsf{ID}^*, b} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ID}^*, \mathbf{m}_b)$ and sends $\mathsf{CT}^*_{\mathsf{ID}^*, b}$ to $\mathcal{A}$.
4. **Phase 2:** same as in Phase 1 except that $\mathcal{A}$ cannot query $\mathcal{O}^{\mathsf{Ext}}$ and $\mathcal{O}^{\mathsf{Td}_\alpha}$ for the identity ID^*.

5. **Guess:** $\mathcal{A}$ output b'.

The adversary $\mathcal{A}$ wins the above game if $b = b'$ and the advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}_{\mathcal{A},\text{IBEET-FA}}^{\text{IND-ID-CPA},Type-\alpha} := \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

2.2 Lattices

We mainly focus on integer lattices, namely discrete subgroups of $\mathbb{Z}^m$. Specially, a lattice Λ in $\mathbb{Z}^m$ with basis $B = [\mathbf{b}_1, \cdots, \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$, where each $\mathbf{b}_i$ is written in column form, is defined as

$$\Lambda := \left\{ \sum_{i=1}^{n} \mathbf{b}_i x_i \middle| x_i \in \mathbb{Z} \ \forall i = 1, \cdots, n \right\} \subseteq \mathbb{Z}^m.$$

We call n the rank of Λ and if $n = m$ we say that Λ is a full rank lattice. In this paper, we mainly consider full rank lattices containing $q\mathbb{Z}^m$, called q-ary lattices, defined as the following, for a given matrix $A \in \mathbb{Z}^{n \times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$

$$\Lambda_q(A) := \left\{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ where } A^{\mathrm{T}} \mathbf{s} = \mathbf{e} \mod q \right\}$$

$$\Lambda_q^{\perp}(A) := \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } A\mathbf{e} = 0 \mod q \}$$

$$\Lambda_q^{\mathbf{u}}(A) := \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } A\mathbf{e} = \mathbf{u} \mod q \}$$

Note that if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(A)$ then $\Lambda_q^{\mathbf{u}}(A) = \Lambda_q^{\perp}(A) + \mathbf{t}$. Hence, one can see $\Lambda_q^{\mathbf{u}}(A)$ as a shift of $\Lambda_q^{\perp}(A)$.

Let $S = \{ \mathbf{s}_1, \cdots, \mathbf{s}_k \}$ be a set of vectors in $\mathbb{R}^m$. We denote by $\|S\| := \max_i \|\mathbf{s}_i\|$ for $i = 1, \cdots, k$, the maximum l_2 length of the vectors in S. We also denote $\tilde{S} := \{ \tilde{\mathbf{s}}_1, \cdots, \tilde{\mathbf{s}}_k \}$ the Gram-Schmidt orthogonalization of the vectors $\mathbf{s}_1, \cdots, \mathbf{s}_k$ in that order. We refer to $\|\tilde{S}\|$ the Gram-Schmidt norm of S.

Ajtai [2] first proposed how to sample a uniform matrix $A \in \mathbb{Z}_q^{n \times m}$ with an associated basis S_A of $\Lambda_q^{\perp}(A)$ with low Gram-Schmidt norm. It is improved later by Alwen and Peikert [3] in the following Theorem.

Theorem 1. *Let $q \geq 3$ be odd and $m := \lceil 6n \log q \rceil$. There is a probabilistic polynomial-time algorithm $\mathsf{TrapGen}(q, n)$ that outputs a pair $(A \in \mathbb{Z}_q^{n \times m}, S \in \mathbb{Z}^{m \times m})$ such that A is statistically close to a uniform matrix in $\mathbb{Z}_q^{n \times m}$ and S is a basis for $\Lambda_q^{\perp}(A)$ satisfying*

$$\|\tilde{S}\| \leq O(\sqrt{n \log q}) \quad and \quad \|S\| \leq O(n \log q)$$

with all but negligible probability in n.

Definition 1 (Gaussian distribution). *Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice. For a vector $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter $\sigma \in \mathbb{R}$, define:*

$$\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(\pi \frac{\|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2} \right) \quad and \quad \rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x}).$$

The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ is

$$\forall \mathbf{y} \in \Lambda \quad , \quad \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}.$$

For convenience, we will denote by ρ_σ and $\mathcal{D}_{\Lambda.\sigma}$ for $\rho_{\mathbf{0},\sigma}$ and $\mathcal{D}_{\Lambda,\sigma,\mathbf{0}}$ respectively. When $\sigma = 1$ we will write ρ instead of ρ_1.

We recall below in Theorem 2 some useful results. The first one comes from [14, Lemma 4.4] . The second one is from [5] and formulated in [1, Theorem 17] and the last one is from [1, Theorem 19].

Theorem 2. *Let $q > 2$ and let A, B be matrices in $\mathbb{Z}_q^{n \times m}$ with $m > n$ and B is rank n. Let T_A, T_B be a basis for $\Lambda_q^\perp(A)$ and $\Lambda_q^\perp(B)$ respectively. Then for $c \in \mathbb{R}^m$ and $U \in \mathbb{Z}_q^{n \times t}$:*

1. *Let M be a matrix in $\mathbb{Z}_q^{n \times m_1}$ and $\sigma \geq \|\widetilde{T_A}\|\omega(\sqrt{\log(m + m_1)})$. Then there exists a PPT algorithm $\mathsf{SampleLeft}(A, M, T_A, U, \sigma)$ that outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^\mathbf{u}(F_1),\sigma}$ where $F_1 := (A \mid M)$. In particular $\mathbf{e} \in \Lambda_q^U(F_1)$, i.e., $F_1 \cdot \mathbf{e} = U \mod q$. In addition, if A is rank n then there is a PPT algorithm $\mathsf{SampleBasisLeft}(A, M, T_A, \sigma)$ that outputs a basis of $\Lambda_q^\perp(F_1)$.*

2. *Let R be a matrix in $\mathbb{Z}^{k \times m}$ and let $s_R := \sup_{\|\mathbf{x}\|=1} \|R\mathbf{x}\|$. Let $F_2 := (A \mid AR + B)$. Then for $\sigma \geq \|\widetilde{T_B}\|s_R\omega(\sqrt{\log m})$, there exists a PPT algorithm $\mathsf{SampleRight}(A, B, R, T_B, U, \sigma)$ that outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+k}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^U(F_2),\sigma}$. In particular $\mathbf{e} \in \Lambda_q^\mathbf{u}(F_2)$, i.e., $F_2 \cdot \mathbf{e} = U \mod q$. In addition, if B is rank n then there is a PPT algorithm $\mathsf{SampleBasisRight}(A, B, R, T_B, \sigma)$ that outputs a basis of $\Lambda_q^\perp(F_2)$.*
   Note that when R is a random matrix in $\{-1,1\}^{m \times m}$ then $s_R < O(\sqrt{m})$ with overwhelming probability (cf. [1, Lemma 15]).

The security of our construction reduces to the LWE (Learning With Errors) problem introduced by Regev [15].

Definition 2 (LWE problem). *Consider publicly a prime q, a positive integer n, and a distribution χ over $\mathbb{Z}_q$. An $(\mathbb{Z}_q, n, \chi)$-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being either a noisy pseudorandom sampler $\mathcal{O}_\mathbf{s}$ associated with a secret $\mathbf{s} \in \mathbb{Z}_q^n$, or a truly random sampler $\mathcal{O}_\$$ who behaviors are as follows:*

$\mathcal{O}_\mathbf{s}$: *samples of the form $(\mathbf{u}_i, v_i) = (\mathbf{u}_i, \mathbf{u}_i^T\mathbf{s} + x_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniform secret key, $\mathbf{u}_i \in \mathbb{Z}_q^n$ is uniform and $x_i \in \mathbb{Z}_q$ is a noise withdrawn from χ.*
$\mathcal{O}_\$$: *samples are uniform pairs in $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

The $(\mathbb{Z}_q, n, \chi)$-LWE problem allows responds queries to the challenge oracle $\mathcal{O}$. We say that an algorithm $\mathcal{A}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{LWE}} := \left| \Pr[\mathcal{A}^{\mathcal{O}_\mathbf{s}} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1] \right|$$

is non-negligible for a random $\mathbf{s} \in \mathbb{Z}_q^n$.

Regev [15] showed that (see Theorem 3 below) when χ is the distribution $\overline{\Psi}_\alpha$ of the random variable $\lfloor qX \rceil \mod q$ where $\alpha \in (0,1)$ and X is a normal random variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ then the LWE problem is hard.

Theorem 3. *If there exists an efficient, possibly quantum, algorithm for deciding the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem for $q > 2\sqrt{n}/\alpha$ then there is an efficient quantum algorithm for approximating the SIVP and GapSVP problems, to within $\tilde{\mathcal{O}}(n/\alpha)$ factors in the l_2 norm, in the worst case.*

Hence if we assume the hardness of approximating the SIVP and GapSVP problems in lattices of dimension n to within polynomial (in n) factors, then it follows from Theorem 3 that deciding the LWE problem is hard when n/α is a polynomial in n.

3   Proposed Construction: IBEET-FA over Integer Lattice

3.1   Construction

Setup(λ) : On input a security parameter λ, set the parameters q, n, m, σ, α as in section 3.3

1. Use $\mathsf{TrapGen}(q, n)$ to generate uniformly random $n \times m$-matrices $A, A' \in \mathbb{Z}_q^{n \times m}$ together with trapdoors $T_A, T_{A'} \in \mathbb{Z}_q^{m \times m}$ respectively.
2. Select $\ell + 1$ uniformly random $n \times m$ matrices $A_1, \cdots, A_\ell, B \in \mathbb{Z}_q^{n \times m}$ where ℓ is the bit length of identity vector ID.
3. Select a uniformly random matrix $U \in \mathbb{Z}_q^{n \times t}$ where t is the bit length of message $\mathbf{m}$.
4. $H : \{0,1\}^* \to \{0,1\}^t$ is a collision-resistant hash function.
5. $H' : \{0,1\}^* \to \{0,1\}^\lambda$ is a collision-resistant hash function.
6. Output the public key and the master secret key

$$\mathsf{PP} = (A, A', A_1, \cdots, A_\ell, B, U) \quad , \quad \mathsf{MSK} = (T_A, T_{A'}).$$

Extract$(\mathsf{PP}, \mathsf{MSK}, \mathsf{ID})$: On input the public parameter PP, a master secret key MSK and an identity $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell) \in \{-1, 1\}^\ell$:

1. Let $A_{\mathsf{ID}} = B + \sum_{i=1}^\ell \mathsf{id}_i A_i \in \mathbb{Z}_q^{n \times m}$.
2. Set $F_{\mathsf{ID}} = (A | A_{\mathsf{ID}}), F'_{\mathsf{ID}} = (A' | A_{\mathsf{ID}}) \in \mathbb{Z}_q^{n \times 2m}$.
3. Sample $E_{\mathsf{ID}}, E'_{\mathsf{ID}} \in \mathbb{Z}^{2m \times 2m}$ as

$$E_{\mathsf{ID}} \leftarrow \mathsf{SampleBasisLeft}(A, A_{\mathsf{ID}}, T_A, \sigma),$$

$$E'_{\mathsf{ID}} \leftarrow \mathsf{SampleBasisLeft}(A', A_{\mathsf{ID}}, T_{A'}, \sigma).$$

4. Output $\mathsf{SK}_{\mathsf{ID}} := (E_{\mathsf{ID}}, E'_{\mathsf{ID}})$.

Note that $F_{\mathsf{ID}} \cdot E_{\mathsf{ID}} = 0, F'_{\mathsf{ID}} \cdot E'_{\mathsf{ID}} = 0$ in $\mathbb{Z}_q$ and $E_{\mathsf{ID}}, E'_{\mathsf{ID}}$ are basis of $\Lambda_q^\perp(F_{\mathsf{ID}})$, $\Lambda_q^\perp(F'_{\mathsf{ID}})$ respectively.

Encrypt$(\mathsf{PP}, \mathsf{ID}, \mathbf{m})$: On input the public parameter PP, an identity $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell) \in \{-1, 1\}^\ell$ and a message $\mathbf{m} \in \{0, 1\}^t$, do:

1. Choose uniformly random $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^n$.
2. Choose $\mathbf{x}_1, \mathbf{x}_2 \in \overline{\Psi}_\alpha^t$ and compute $\mathbf{c}_1 = U^T \mathbf{s}_1 + \mathbf{x}_1 + \mathbf{m} \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$, $\mathbf{c}_2 = U^T \mathbf{s}_2 + \mathbf{x}_2 + H(\mathbf{m}) \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$.
3. Pick ℓ uniformly random matrices $R_i \in \{-1, 1\}^{m \times m}$ for $i = 1, \cdots, \ell$, set

$$R_{\mathsf{ID}} = \sum_{i=1}^{\ell} \mathsf{id}_i R_i \in \{-\ell, \cdots, \ell\}^{m \times m}.$$

4. Choose a uniformly at random $R \in \{-\ell, \cdots, \ell\}^{m \times m}$ and set

$$F_1 = (F_{\mathsf{ID}} | AR) \quad, \quad F_2 = (F'_{\mathsf{ID}} | AR) \in \mathbb{Z}_q^{n \times 3m}.$$

5. Choose $\mathbf{y}_1, \mathbf{y}_2 \in \overline{\Psi}_\alpha^m$ and set

$$\mathbf{z}_1 = R_{\mathsf{ID}}^T \mathbf{y}_1 \quad, \quad \mathbf{z}_2 = R_{\mathsf{ID}}^T \mathbf{y}_2 \in \mathbb{Z}_q^m,$$

$$\mathbf{r}_1 = R^T \mathbf{y}_1 \quad, \quad \mathbf{r}_2 = R^T \mathbf{y}_2 \in \mathbb{Z}_q^m.$$

6. Compute $\mathbf{c}_3 = F_1^T \mathbf{s}_1 + [\mathbf{y}_1 | \mathbf{z}_1 | \mathbf{r}_1]^T \in \mathbb{Z}_q^{3m}$, $\mathbf{c}_4 = F_2^T \mathbf{s}_2 + [\mathbf{y}_2 | \mathbf{z}_2 | \mathbf{r}_2]^T \in \mathbb{Z}_q^{3m}$.
7. Compute $\mathbf{c}_5 = H'(R || \mathbf{c}_1 || \mathbf{c}_2 || \mathbf{c}_3 || \mathbf{c}_4)$.
8. The ciphertext is

$$\mathsf{CT}_{\mathsf{ID}} = (R, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5) \in \mathbb{Z}_q^{m \times m} \times \mathbb{Z}_q^{2t+6m} \times \{0, 1\}^\lambda.$$

Decrypt$(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{CT})$: On input public parameter PP, private key $\mathsf{SK}_{\mathsf{ID}} = (E_{\mathsf{ID}}, E'_{\mathsf{ID}})$ and a ciphertext $\mathsf{CT} = (R, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5)$, do:

1. Check if $\mathbf{c}_5 = H'(R || \mathbf{c}_1 || \mathbf{c}_2 || \mathbf{c}_3 || \mathbf{c}_4)$. If not, returns $\bot$.
2. Sample $\mathbf{e} \in \mathbb{Z}_q^{3m \times t}$ as

$$\mathbf{e} \leftarrow \mathsf{SampleLeft}(F_{\mathsf{ID}}, AR, E_{\mathsf{ID}}, U, \sigma).$$

3. Note that $F_1 \cdot \mathbf{e} = U \in \mathbb{Z}_q^{n \times t}$. We compute $\mathbf{w} \leftarrow \mathbf{c}_1 - \mathbf{e}^T \mathbf{c}_3 \in \mathbb{Z}_q^t$.
4. For each $i = 1, \cdots, t$, compare w_i and $\lfloor \frac{q}{2} \rfloor$. If they are close, i.e. $\left| w_i - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$, output $m_i = 1$, otherwise $m_i = 0$. We then obtain the message $\mathbf{m}$.
5. Sample $\mathbf{e}' \in \mathbb{Z}_q^{3m \times t}$ as

$$\mathbf{e}' \leftarrow \mathsf{SampleLeft}(F'_{\mathsf{ID}}, AR, E'_{\mathsf{ID}}, U, \sigma).$$

6. Note that $F_2 \cdot \mathbf{e}' = U \in \mathbb{Z}_q^{n \times t}$. We compute $\mathbf{w}' \leftarrow \mathbf{c}_2 - (\mathbf{e}')^T \mathbf{c}_4 \in \mathbb{Z}_q^t$.
7. For each $i = 1, \cdots, t$, compare w'_i and $\lfloor \frac{q}{2} \rfloor$. If they are close, i.e. $\left| w_i - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$, output $h_i = 1$, otherwise $h_i = 0$. We then obtain the vector $\mathbf{h}$.
8. Finally, if $\mathbf{h} = H(\mathbf{m})$ then output $\mathbf{m}$, otherwise output $\bot$.

Flexible Authorization: Suppose we have two identities in the system ID_i and ID_j along with $\mathsf{CT}_{\mathsf{ID}_i} = (R_i, \mathbf{c}_{i1}, \mathbf{c}_{i2}, \mathbf{c}_{i3}, \mathbf{c}_{i4}, \mathbf{c}_{i5})$ and $\mathsf{CT}_{\mathsf{ID}_j} = (R_j, \mathbf{c}_{j1}, \mathbf{c}_{j2}, \mathbf{c}_{j3}, \mathbf{c}_{j4}, \mathbf{c}_{j5})$ be their ciphertexts respectively.

Type-1 Authorization :
- $\mathbf{Td}_1(\mathsf{SK}_{\mathsf{ID}})$: On input $\mathsf{SK}_{\mathsf{ID}} = (E_{\mathsf{ID}}, E'_{\mathsf{ID}})$, outputs a trapdoor $\mathsf{td}_{1,\mathsf{ID}} = E'_{\mathsf{ID}}$.
- $\mathbf{Test}(\mathsf{td}_{1,\mathsf{ID}_i}, \mathsf{td}_{1,\mathsf{ID}_j}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j})$: On input trapdoors $\mathsf{td}_{1,\mathsf{ID}_i}, \mathsf{td}_{1,\mathsf{ID}_j}$ and ciphertexts $\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j}$ for identities $\mathsf{ID}_i, \mathsf{ID}_j$ respectively, do
  1. For each identity ID_i (resp. ID_j), do the following:
     - Check if $\mathbf{c}_{i5} = H'(R_i||\mathbf{c}_{i1}||\mathbf{c}_{i2}||\mathbf{c}_{i3}||\mathbf{c}_{i4})$. If not, returns $\perp$.
     - Take R_i from $\mathsf{CT}_{\mathsf{ID}_i}$ and sample $\mathbf{e}'_i \in \mathbb{Z}_q^{3m \times t}$ as

     $$\mathbf{e}'_i \leftarrow \mathsf{SampleLeft}(F'_{\mathsf{ID}}, AR_i, E'_{\mathsf{ID}}, U, \sigma)$$

     Note that $F_{i2} \cdot \mathbf{e}'_i = U \in \mathbb{Z}_q^{n \times t}$.
     - Compute $\mathbf{w}'_\mathbf{i} \leftarrow \mathbf{c}_{i2} - (\mathbf{e}'_i)^T \mathbf{c}_{i4} \in \mathbb{Z}_q^t$. For each $k = 1, \cdots, t$, compare w_{ik} with $\lfloor \frac{q}{2} \rfloor$ and output $\mathbf{h}_{ik} = 1$ if they are close, and 0 otherwise. We obtain the vector $\mathbf{h}_i$ (resp. $\mathbf{h}_j$).
  2. Output 1 if $\mathbf{h}_i = \mathbf{h}_j$ and 0 otherwise.

Type-2 Authorization :
- $\mathbf{Td}_2(\mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}_{\mathsf{ID}})$: On input $\mathsf{SK}_{\mathsf{ID}} = (E_{\mathsf{ID}}, E'_{\mathsf{ID}})$ and its ciphertext $\mathsf{CT}_{\mathsf{ID}}$, outputs a trapdoor $\mathsf{td}_{2,\mathsf{ID}}$ with following procedure:
  1. Check if $\mathbf{c}_5 = H'(R||\mathbf{c}_1||\mathbf{c}_2||\mathbf{c}_3||\mathbf{c}_4)$. If not, returns $\perp$.
  2. Take R from $\mathsf{CT}_{\mathsf{ID}}$ and sample $\mathsf{td}_{2,\mathsf{ID}} = \mathbf{e}' \in \mathbb{Z}_q^{3m \times t}$ as

  $$\mathbf{e}' \leftarrow \mathsf{SampleLeft}(F'_{\mathsf{ID}}, AR, E'_{\mathsf{ID}}, U, \sigma).$$

  Note that $F_2 \cdot \mathbf{e}' = U \in \mathbb{Z}_q^{n \times t}$
- $\mathbf{Test}(\mathsf{td}_{2,\mathsf{ID}_i}, \mathsf{td}_{2,\mathsf{ID}_j}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j})$: On input trapdoors $\mathsf{td}_{2,\mathsf{ID}_i}, \mathsf{td}_{2,\mathsf{ID}_j}$ and ciphertexts $\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j}$ for identities $\mathsf{ID}_i, \mathsf{ID}_j$ respectively, do
  1. Compute $\mathbf{w}'_\mathbf{i} \leftarrow \mathbf{c}_{i2} - (\mathbf{e}'_i)^T \mathbf{c}_{i4} \in \mathbb{Z}_q^t$. For each $k = 1, \cdots, t$, compare w_{ik} with $\lfloor \frac{q}{2} \rfloor$ and output $\mathbf{h}_{ik} = 1$ if they are close, and 0 otherwise. We then obtain the vector $\mathbf{h}_i$ (resp. $\mathbf{h}_j$).
  2. Output 1 if $\mathbf{h}_i = \mathbf{h}_j$ and 0 otherwise.

Type-3 Authorization :
- $\mathbf{Td}_3(\mathsf{SK}_{\mathsf{ID}_i})$: On input $\mathsf{SK}_{\mathsf{ID}_i} = (E_{\mathsf{ID}_i}, E'_{\mathsf{ID}_i})$, returns $\mathsf{td}_{3,\mathsf{ID}_i} = E'_{\mathsf{ID}_i}$.
- $\mathbf{Td}_3(\mathsf{SK}_{\mathsf{ID}_j}, \mathsf{CT}_j)$: On input an identity's secret key $\mathsf{SK}_{\mathsf{ID}_j} = (E_{\mathsf{ID}_j}, E'_{\mathsf{ID}_j})$ and ciphertext CT_j, it outputs a trapdoor $\mathsf{td}_{3,\mathsf{ID}_j}$ using procedure:
  1. Check if $\mathbf{c}_{j5} = H'(R_j||\mathbf{c}_{j1}||\mathbf{c}_{j2}||\mathbf{c}_{j3}||\mathbf{c}_{j4})$. If not, returns $\perp$.
  2. Take R_j from $\mathsf{CT}_{\mathsf{ID}_j}$ and sample $\mathsf{td}_{3,\mathsf{ID}_j} = \mathbf{e}'_j \in \mathbb{Z}_q^{3m \times t}$ as

  $$\mathbf{e}'_j \leftarrow \mathsf{SampleLeft}(F'_{\mathsf{ID}}, AR_j, E'_{\mathsf{ID}}, U, \sigma).$$

  Note that $F_{j2} \cdot \mathbf{e}'_j = U \in \mathbb{Z}_q^{n \times t}$.
- $\mathbf{Test}(\mathsf{td}_{3,\mathsf{ID}_i}, \mathsf{td}_{3,\mathsf{ID}_j}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j})$: On input trapdoors $\mathsf{td}_{3,\mathsf{ID}_i}, \mathsf{td}_{3,\mathsf{ID}_j}$ and ciphertexts $\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_j}$ for identities $\mathsf{ID}_i, \mathsf{ID}_j$ respectively, do
  1. For identity ID_i, do the following:
     - Check if $\mathbf{c}_{i5} = H'(R_i||\mathbf{c}_{i1}||\mathbf{c}_{i2}||\mathbf{c}_{i3}||\mathbf{c}_{i4})$. If not, returns $\perp$.
     - Take R_i from $\mathsf{CT}_{\mathsf{ID}_i}$ and sample $\mathbf{e}'_i \in \mathbb{Z}_q^{3m \times t}$ as

     $$\mathbf{e}'_i \leftarrow \mathsf{SampleLeft}(F'_{\mathsf{ID}}, AR_i, E'_{\mathsf{ID}}, U, \sigma).$$

     Note that $F_{i2} \cdot \mathbf{e}'_i = U \in \mathbb{Z}_q^{n \times t}$.

- Compute $\mathbf{w'_i} \leftarrow \mathbf{c}_{i2} - (\mathbf{e}'_i)^T \mathbf{c}_{i4} \in \mathbb{Z}_q^t$. For each $k = 1, \cdots, t$, compare w_{ik} with $\lfloor \frac{q}{2} \rfloor$ and output $\mathbf{h}_{ik} = 1$ if they are close, and 0 otherwise. We then obtain the vector $\mathbf{h}_i$.
  2. For each ID_j: Compute $\mathbf{w'_j} \leftarrow \mathbf{c}_{j2} - (\mathbf{e}'_j)^T \mathbf{c}_{j4} \in \mathbb{Z}_q^t$. For each $k = 1, \cdots, t$, compare w_{ik} with $\lfloor \frac{q}{2} \rfloor$ and output $\mathbf{h}_{jk} = 1$ if they are close, and 0 otherwise. We then obtain the vector $\mathbf{h}_j$.
  3. Output 1 if $\mathbf{h}_i = \mathbf{h}_j$ and 0 otherwise.

Theorem 4. *Proposed IBEET-FA construction above is correct if H is a collision-resistant hash function.*

Proof. We have that if CT is a valid ciphertext of $\mathbf{m}$ then the decryption will always output $\mathbf{m}$. Furthermore, Test of Type-α (for $\alpha = 1, 2, 3$) checks whether $H(\mathbf{m}_i) = H(\mathbf{m}_j)$ where CT_i and CT_j are valid ciphertexts of two identities ID_i and ID_j respectively. It outputs 1, meaning that $\mathbf{m}_i = \mathbf{m}_j$, which is always correct with overwhelming probability since H is collision resistant. Hence, the proposed IBEET-FA described above is correct.

3.2   Security analysis

We show that the our proposed IBEET-FA construction is OW-ID-CPA secure against Type-I adversaries (cf. Theorem 5) and IND-ID-CPA secure against Type-II adversaries (cf. Theorem 6). Due to space limitation, we prove in details Theorem 5, while we refer the detailed proof for Theorem 6 (which is simpler than that for Theorem 5) to the full version [24]

Theorem 5. *The IBEET-FA system with parameters $(q, n, m, \sigma, \alpha)$ as in Section (3.3) is OW-ID-CPA secure provided that H is a one-way hash function and the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumption holds.*

Proof. Assume that there is a Type-I adversary $\mathcal{A}$ who breaks the OW-ID-CPA security of the IBEET-FA scheme. We construct an algorithm $\mathcal{B}$ who solves the LWE problem using $\mathcal{A}$. The proof proceeds in a sequence of games where the first game is identical to the OW-ID-CPA.

Game 0. This is the original OW-ID-CPA game between the attacker $\mathcal{A}$ against the scheme and the OW-ID-CPA challenger. Recall that in this Game 0 the challenger generates public parameter PP by choosing $\ell + 3$ random matrices $A, A', A_1, \cdots, A_\ell, B$ in $\mathbb{Z}_q^{n \times m}$. Let $R_i^* \in \{-1, 1\}^{m \times m}$ for $i = 1, \cdots, \ell$ be the ephemeral random matrices generated when creating the ciphertext $\mathsf{CT}^*_{\mathsf{ID}^*}$.

Game 1. At setup phase, the challenger $\mathcal{B}$ chooses ℓ uniform random matrices R_i^*, and ℓ random scalars $h_i \in \mathbb{Z}_q$ for $i = 1, \cdots, \ell$. After that, it generates A, A', B as in Game 0 and constructs the matrices A_i for $i = 1, \cdots, \ell$ as

$$A_i \leftarrow A \cdot R_i^* - h_i \cdot B \in \mathbb{Z}_q^{n \times m}.$$

The remainder of Game 1 is unchanged with R_i^*, $i = 1, \cdots, \ell$ is used to generate the challenge ciphertext. Note that these $R_i^* \in \{-1, 1\}^{m \times m}$ are chosen in advance in setup phase without the knowledge of ID^*. Using [1, Lemma 13], we prove that the matrices A_i are statistically close to uniform random independent. Therefore, the adversary's view in Game 1 is identical to that in Game 0.

Game 2. We add an abort that is independent of adversary's view as follow:
  - In the setup phase, the challenger chooses random $h_i \in \mathbb{Z}_q$, $i = 1, \cdots, \ell$ and keeps these values private.
  - In the final guess phase, the adversary outputs a guest $\mathbf{m}'$ for $\mathbf{m}$. The challenger now does the following:
    1. **Abort check:** for all queries $\mathsf{SK_{ID}}$ to the extract secret key oracle $\mathcal{O}^{\mathsf{Ext}}$, the challenger checks whether the identity $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell)$ satisfies $1 + \sum_{i=1}^{\ell} \mathsf{id}_i h_i \neq 0$ and $1 + \sum_{i=1}^{\ell} \mathsf{id}_i^* h_i = 0$. If not then $\mathcal{B}$ overwrites $\mathbf{m}'$ with a fresh random message and aborts the game. Note that this is unnoticed from the adversary's view and $\mathcal{B}$ can even abort the game as soon as the condition is true.
    2. **Artificial abort:** $\mathcal{B}$ samples a message Γ such that $\Pr[\Gamma = 1] = \mathcal{G}$(all queries $\mathsf{SK_{ID}}$) where $\mathcal{G}$ is defined in [1, Lemma 28]. If $\Gamma = 1$, $\mathcal{B}$ overwrites $\mathbf{m}'$ with a fresh message and make a artificial abort.
  This completes the description of Game 2.

Game 3. We choose A is a uniform random matrix in $\mathbb{Z}_q^{n \times m}$. However, matrices A', B and their trapdoor $T_{A'}, T_B$ are generated through $\mathsf{TrapGen}(q, n)$, where $T_{A'}, T_B$ are basis of $\Lambda_q^\perp(A'), \Lambda_q^\perp(B)$ respectively. The construction of A_i for $i = 1, \cdots, \ell$ remains the same: $A_i = AR_i^* - h_i B$. When $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{Ext}}(\mathsf{ID})$ where $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell)$, the challenger generate secret key using trapdoor T_B as follows:
  - Challenger $\mathcal{B}$ sets $F_{\mathsf{ID}} := (A|B + \sum_{i=1}^{\ell} \mathsf{id}_i A_i) = (A|AR_{\mathsf{ID}} + h_{\mathsf{ID}} B)$, where

$$R_{\mathsf{ID}} \leftarrow \sum_{i=1}^{\ell} \mathsf{id}_i R_i^* \in \mathbb{Z}_q^{m \times m} \quad \text{and} \quad h_{\mathsf{ID}} \leftarrow 1 + \sum_{i=1}^{\ell} \mathsf{id}_i h_i \in \mathbb{Z}_q. \quad (1)$$

  - If $h_{\mathsf{ID}} = 0$ then abort the game and pretend that the adversary outputs a random message $\mathbf{m}'$ as in Game 2.
  - Sample $E_{\mathsf{ID}} \leftarrow \mathsf{SampleBasisRight}(A, h_{\mathsf{ID}} B, R_{\mathsf{ID}}, T_B, \sigma)$. Since h_{ID} is non-zero, T_B is also a trapdoor for $h_{\mathsf{ID}} B$. Hence the output E_{ID} satisfies $F_{\mathsf{ID}} \cdot E_{\mathsf{ID}} = 0$ in $\mathbb{Z}_q$ and E_{ID} is basis of $\Lambda_q^\perp(F_{\mathsf{ID}})$. Moreover, Theorem 2 shows that when $\sigma > \|\widetilde{T_B}\| s_{R_{\mathsf{ID}}} \omega(\sqrt{\log m})$ with $s_{R_{\mathsf{ID}}} := \|R_{\mathsf{ID}}\|$
  - Finally $\mathcal{B}$ return $\mathsf{SK} = (E_{\mathsf{ID}}, E'_{\mathsf{ID}})$ to $\mathcal{A}$ where E'_{ID} is computed as in the real construction $E'_{\mathsf{ID}} \leftarrow \mathsf{SampleBasisLeft}(A', A_{\mathsf{ID}}, T_{A'}, \sigma)$.
  Note that when $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{Td}_\alpha}$ (including ID^*), the trapdoor is still computed as in the real construction.
  The rest of Game 3 is similar to Game 2. In particular, $\mathcal{B}$ uses abort check in challenge phase and artificial abort in guess phase. Then Game 3 and Game

2 are identical in the adversary's view.

Game 4. Game 4 is identical to Game 3, except that the challenge ciphertext is always chosen as a random independent element. And thus $\mathcal{A}$'s advantage is always 0. The remaining part is to show Game 3 and Game 4 are computationally indistinguishable. If the abort event happens then the games are clearly indistinguishable. Hence, we only focus on sequences of queries that do not cause an abort.

Reduction from LWE. Recall that an LWE problem instance is provided as a sampling oracle $\mathcal{O}$ that can be either truly random $\mathcal{O}_\$$ or a noisy pseudo-random $\mathcal{O}_s$ for some secret random $s \in \mathbb{Z}_q^n$. Suppose now $\mathcal{A}$ has a non-negligible advantage in distinguishing Game 3 and Game 4, we use $\mathcal{A}$ to construct $\mathcal{B}$ to solve the LWE problem as follows.

Instance. First of all, $\mathcal{B}$ requests from $\mathcal{O}$ and receives, for each $j = 1, \cdots, t$ a fresh pair of $(\mathbf{u}_i, d_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ and for each $i = 1, \cdots, m$, a fresh pair of $(\mathbf{a}_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Setup. $\mathcal{B}$ constructs the public parameter PP as follows:

1. Assemble the random matrix $A \in \mathbb{Z}_q^{n \times m}$ from m of previously given LWE samples by letting the i-th column of A to be the n-vector $\mathbf{a}_i$ for all $i = 1, \cdots, m$.
2. Assemble the first t unused LWE samples $\mathbf{u}_1, \cdots, \mathbf{u}_t$ to become a public random matrix $U \in \mathbb{Z}_q^{n \times t}$.
3. Matrix A', B and trapdoor $T_{A'}, T_B$ are generated as in Game 4 while matrices A_i for $i = 1, \cdots, \ell$ are constructed as in Game 1.
4. Set $\mathsf{PP} := (A, A', A_1, \cdots, A_\ell, B, U)$ and send to $\mathcal{A}$.

Queries. $\mathcal{B}$ answers the queries as in Game 4, including aborting the game.

Challenge. Now when $\mathcal{A}$ send $\mathcal{B}$ a target identity $\mathsf{ID}^* = (\mathsf{id}_1^*, \cdots, \mathsf{id}_\ell^*)$. $\mathcal{B}$ then choose a random message $\mathbf{m}^*$ and computes the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*}^* = (R^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*, \mathbf{c}_4^*, \mathbf{c}_5^*)$ as follows:

1. Assemble $\mathbf{d}^* = [d_1, \cdots, d_t]^T \in \mathbb{Z}_q^t$, set $\mathbf{c}_1^* \leftarrow \mathbf{d}^* + \mathbf{m}^* \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$.
2. Choose uniformly random $\mathbf{s}_2 \in \mathbb{Z}_q^n$ and $\mathbf{x}_2 \leftarrow \overline{\Psi}_\alpha^t$, compute

$$\mathbf{c}_2^* = U^T \mathbf{s}_2 + \mathbf{x}_2 + H(\mathbf{m}^*) \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t.$$

3. Compute $R_{\mathsf{ID}^*}^* := \sum_{i=1}^\ell \mathsf{id}_i^* R_i^* \in \{-\ell, \cdots, \ell\}^{m \times m}$.
4. Choose uniformly at random $R^* \in \{-\ell, \cdots, \ell\}^{m \times m}$.
5. Assemble $\mathbf{v}^* = [v_1, \cdots, v_m]^T \in \mathbb{Z}_q^m$. Set $\mathbf{c}_3^* := \begin{bmatrix} \mathbf{v}^* \\ (R_{\mathsf{ID}^*}^*)^T \mathbf{v}^* \\ (R^*)^T \mathbf{v}^* \end{bmatrix} \in \mathbb{Z}_q^{3m}$.
6. Choose $\mathbf{y}_2 \leftarrow \overline{\Psi}_\alpha^m$ and set $\mathbf{c}_4^* := \begin{bmatrix} A'^T \mathbf{s}_2 + \mathbf{y}_2 \\ (AR_{\mathsf{ID}^*}^*)^T \mathbf{s}_2 + (R_{\mathsf{ID}^*}^*)^T \mathbf{y}_2 \\ (AR^*)^T \mathbf{s}_2 + (R^*)^T \mathbf{y}_2 \end{bmatrix} \in \mathbb{Z}_q^{3m}$.
7. Compute $\mathbf{c}_5^* = H'(R^* || \mathbf{c}_1^* || \mathbf{c}_2^* || \mathbf{c}_3^* || \mathbf{c}_4^*)$.

Then $\mathcal{B}$ sends $\mathsf{CT}^*_{\mathsf{ID}^*} = (R^*, \mathbf{c}^*_1, \mathbf{c}^*_2, \mathbf{c}^*_3, \mathbf{c}^*_4, \mathbf{c}^*_5)$ to $\mathcal{A}$.

We argue that when the LWE oracle is pseudorandom (i.e. $\mathcal{O} = \mathcal{O}_s$) then $\mathsf{CT}^*_{\mathsf{ID}^*}$ is distributed exactly as in Game 4. It suffices to argue only in case of no abort. Note that, since one has $h_{\mathsf{ID}^*} = 0$ and so

$$F_{\mathsf{ID}^*} = (A|AR^*_{\mathsf{ID}^*}) \quad , \quad F^*_1 = (A|AR^*_{\mathsf{ID}^*}|AR^*)$$

Because oracle is pseudorandom, then we have $\mathbf{v}^* = A^T\mathbf{s}_1 + \mathbf{y}_1$ for some random noise vector $\mathbf{y}_1 \leftarrow \overline{\Psi}^m_\alpha$ respectively. Therefore, $\mathbf{c}^*_3$ in Step 5 satisfies:

$$\mathbf{c}^*_3 := \begin{bmatrix} A^T\mathbf{s}_1 + \mathbf{y}_1 \\ (R^*_{\mathsf{ID}^*})^T(A^T\mathbf{s}_1 + \mathbf{y}_1) \\ (R^*)^T(A^T\mathbf{s}_1 + \mathbf{y}_1) \end{bmatrix} = (F^*_1)^T\mathbf{s}_1 + \begin{bmatrix} \mathbf{y}_1 \\ (R^*_{\mathsf{ID}^*})^T\mathbf{y}_1 \\ (R^*)^T\mathbf{y}_1 \end{bmatrix}.$$

One can easily see that $\mathbf{c}^*_4 := (F^*_2)^T\mathbf{s}_2 + \begin{bmatrix} \mathbf{y}_2 \\ (R^*_{\mathsf{ID}^*})^T\mathbf{y}_2 \\ (R^*)^T\mathbf{y}_2 \end{bmatrix}.$

Moreover, for some $\mathbf{x}_1, \leftarrow \overline{\Psi}^t_\alpha$, we have $\mathbf{d}^* = U^T\mathbf{s}_1 + \mathbf{x}_1$, therefore

$$\mathbf{c}^*_1 = U^T\mathbf{s}_1 + \mathbf{x}_1 + \mathbf{m}^*\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}^t_q$$

Therefore $\mathsf{CT}^*_{\mathsf{ID}^*}$ is a valid ciphertext.
When $\mathcal{O} = \mathcal{O}_\$$ we have that $\mathbf{d}^*$ is uniform in $\mathbb{Z}^t_q$ and $\mathbf{v}^*$ is uniform in $\mathbb{Z}^m_q$. Then obviously $\mathbf{c}^*_1$ is uniform. It follows also from the leftover hash lemma (cf. [17, Theorem 8.38]) that $\mathbf{c}^*_3$ is also uniform and independence in $\mathbb{Z}^3_q m$. Consequently, the challenge ciphertext $\mathsf{CT}^*_{\mathsf{ID}^*}$ is always uniform in $\mathbb{Z}^{m \times m}_q \times \mathbb{Z}^{2t+6m}_q \times \{0,1\}^\lambda$.

Guess. After making additional queries in Phase 2, $\mathcal{A}$ guesses if it is interacting with Game 3 or Game 4. The simulator also implements the artificial abort for these Games and uses the final guess as the answer to the LWE problem.

We claim that when $\mathcal{O} = \mathcal{O}_\mathbf{s}$ then the adversary's view is as in Game 3. When $\mathcal{O} = \mathcal{O}_\$$ then the view of adversary is as in Game 4. Hence $\mathcal{B}$'s advantage in solving the LWE problem is the same as the advantage of $\mathcal{A}$ in distinguishing Game 3 and Game 4 as required. This completes the description of algorithm $\mathcal{B}$ and completes the proof.

Theorem 6. *The IBEET-FA system with parameters $(q, n, m, \sigma, \alpha)$ as in Section (3.3) is IND-ID-CPA secure provided that H is a one-way hash function and the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumption holds. In particular, suppose there exists a probabilistic algorithm $\mathcal{A}$ that wins the IND-ID-CPA game with advantage ϵ. Then there is a probabilistic algorithm $\mathcal{B}$ that solves the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE problem.*

Proof. The proof is quite similar to and simpler than that of Theorem 5 in which we show our construction is indistinguishable from random, meaning the challenge ciphertext is indistinguishable from a random element in the ciphertext space.

We refer the readers to the full version [24] for a detailed proof.

3.3   Setting Parameters

Following [1, Section 7.3], we choose our parameters satisfying:

- that the TrapGen works, i.e., $m > 6n \log q$.
- that σ is large enough for SampleLeft, SampleRight, SampleBasisLeft and SampleBasisRight to work, i.e., $\sigma > \max\{\|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(2m)}), \|\widetilde{T_B}\| \cdot s_{R_{\mathsf{ID}}} \cdot \omega(\sqrt{\log m})\}$. Note that, when R is a random matrix in $\{-1, 1\}^{m \times m}$ then $s_R < O(\sqrt{m})$ with overwhelming probability (cf. [1, Lemma 15]). Hence when R_{ID} is a random matrix in $\{-\ell, \ell\}^{m \times m}$ then $s_{R_{\mathsf{ID}}} < O(\ell\sqrt{m})$. Also, note that $\|\widetilde{T_A}\|, \|\widetilde{T_B}\| \leq O(\sqrt{n \log q})$.
- that Regev's reduction for the LWE problem to apply, i.e., $q > 2\sqrt{n}/\alpha$.
- that our security reduction applies (i.e., $q > 2Q$ where Q is the number of identity queries from the adversary).
- the error term in decryption is less than $q/5$ with high probability, i.e., $q = \Omega(\sigma m^{3/2})$ and $\alpha < [\sigma l m \omega(\sqrt{\log m})]^{-1}$.

4   Conclusion

In this paper, we propose a direct construction of IBEET-FA based on the hardness of Learning With Errors problem. This construction inherited the efficiency from direct construction of PKEET by Duong et al [7, 8]. However, in order to support flexible authorization, the ciphertext size and secret key size increase. One can improve it by using elegant methods in construction and security proof. We will leave as a future work for improving our schemes to achieve the CCA2-security as well as reducing the storage size. Additionally, as mentioned earlier, the existing security model for IBEET and hence for IBEET-FA is not sufficiently strong in some scenarios. Therefore, finding a stronger security model for IBEET and IBEET-FA would also be an interesting future work.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, pages 553–572, 2010.
2. Miklós Ajtai. Generating hard instances of the short basis problem. In *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, pages 1–9, 1999.

3. Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, pages 75–86, 2009.

4. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

5. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, pages 523–552, 2010.

6. Dung Hoang Duong, Kazuhide Fukushima, Shinsaku Kiyomoto, Partha Sarathi Roy, and Willy Susilo. A lattice based public key encryption with equality test in standard model. In *Information Security and Privacy - 24nd Australasian Conference, ACISP 2019, Christchurch, New Zealand, July 3-5, 2019, Proceedings*, pages 168–183, 2019.

7. Dung Hoang Duong, Huy Quoc Le, Partha Sarathi Roy, and Willy Susilo. Lattice-Based IBE with Equality Test in Standard Model. In *Provable Security*, pages 19-40, 2019.

8. Dung Hoang Duong, Kazuhide Fukushima, Shinsaku Kiyomoto, Partha Sarathi Roy, and Willy Susilo. Lattice-based public key encryption with equality test supporting flexible authorization in standard model. In *ArXiv, 2005.05308*, 2020.

9. Hyung Tae Lee, San Ling, Jae Hong Seo, Huaxiong Wang, and Taek-Young Youn. Public key encryption with equality test in the standard model. *Cryptology ePrint Archive*, Report 2016/1182, 2016.

10. Hyung Tae Lee, San Ling, Jae Hong Seo, and Huaxiong Wang. Semi-generic construction of public key encryption and identity-based encryption with equality test. *Inf. Sci.*, 373:419–440, 2016.

11. Hyung Tae Lee, Huaxiong Wang, and Kai Zhang. Security analysis and modification of id-based encryption with equality test from ACISP 2017. In *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings*, pages 780–786, 2018.

12. Sha Ma. Identity-based encryption with outsourced equality test in cloud computing. *Information Sciences*, 328:389–402, 2016.

13. Sha Ma, Qiong Huang, Mingwu Zhang and Bo Yang. Efficient Public Key Encryption With Equality Test Supporting Flexible Authorization. *IEEE Transactions on Information Forensics and Security*, Volume: 10 , Issue: 3 , March 2015.

14. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 372–381, 2004.

15. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

16. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

17. Victor Shoup. *A computational introduction to number theory and algebra.* Cambridge University Press, second edition, 2008.

18. L. Wu and Y. Zhang and K. R. Choo and D. He. Efficient Identity-Based Encryption Scheme with Equality Test in Smart City. In IEEE Transactions on Sustainable Computing, 3(1): 44-53, 2018.

19. Wu, T., Ma, S., Mu, Y., Zeng, S. Id-based encryption with equality test against insider attack. In: Pieprzyk, J., Suriadi, S. (eds.) Information Security and Privacy. pp. 168–183. Springer International Publishing, Cham (2017).
20. Sha Ma. Identity-based encryption with outsourced equality test in cloud computing. Information Sciences 328, 389–402 (2016).
21. Alornyo, S., Asante, M., Hu, X., Mireku, K.K. Encrypted Traffic Analytic using identity-based Encryption with Equality Test for Cloud Computing. In: 2018 IEEE 7th International Conference on Adaptive Science Technology (ICAST). pp. 1–4 (2018)
22. Ramadan, M., Liao, Y., Li, F., Zhou, S., Abdalla, H. IBEETRSA: identity-based encryption with equality test over RSA for wireless body area networks. MONET 25(1), 223– 233 (2020). https://doi.org/10.1007/s11036-019-01215-9.
23. David Galindo and Ichiro Hasuo, Security Notions for Identity Based Encryption. Cryptology ePrint Archive, Report 2005/253, 2005, https://eprint.iacr.org/2005/253,
24. Giang Linh Duc Nguyen, Willy Susilo, Dung Hoang Duong, Huy Quoc Le and Fuchun Guo, Lattice-based IBE with Equality Test Supporting Flexible Authorization in the Standard Model (full version). https://arxiv.org/abs/2010.14077 (2020).