A Lattice-Based Public Key Encryption with Equality Test in Standard Model

Dung Hoang Duong[1(✉)], Kazuhide Fukushima[2], Shinsaku Kiyomoto[2], Partha Sarathi Roy[2(✉)], and Willy Susilo[1]

[1] Institute of Cybersecurity and Cryptology,
School of Computing and Information Technology, University of Wollongong,
Northfields Avenue, Wollongong, NSW 2522, Australia
{hduong,wsusilo}@uow.edu.au
[2] Information Security Laboratory, KDDI Research, Inc.,
2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
{ka-fukushima,kiyomoto,pa-roy}@kddi-research.jp

Abstract. Public key encryption with equality test (PKEET) allows testing whether two ciphertexts are generated by the same message or not. PKEET is a potential candidate for many practical applications like efficient data management on encrypted databases. Potential applicability of PKEET leads to intensive research from its first instantiation by Yang et al. (CT-RSA 2010). Most of the followup constructions are secure in the random oracle model. Moreover, the security of all the concrete constructions is based on number-theoretic hardness assumptions which are vulnerable in the post-quantum era. Recently, Lee et al. (ePrint 2016) proposed a generic construction of PKEET schemes in the standard model and hence it is possible to yield the first instantiation of PKEET schemes based on lattices. Their method is to use a 2-level hierarchical identity-based encryption (HIBE) scheme together with a one-time signature scheme. In this paper, we propose, for the first time, a direct construction of a PKEET scheme based on the hardness assumption of lattices in the standard model. More specifically, the security of the proposed scheme is reduces to the hardness of the *Learning With Errors* problem. We have used the idea of the full identity-based encryption scheme by Agrawal et al. (EUROCRYPT 2010) to construct the proposed PKEET.

1 Introduction

Public key encryption with equality test (PKEET), which was first introduced by Yang et al. [21], is a special kind of public key encryption that allows anyone with a given trapdoor to test whether two ciphertexts are generated by the same message. This property is of use in various practical applications, such as keyword search on encrypted data, encrypted data partitioning for efficient encrypted data management, personal health record systems, spam filtering in encrypted email systems and so on. Due to its numerous practical applications,

J. Jang-Jaccard and F. Guo (Eds.): ACISP 2019, LNCS 11547, pp. 138–155, 2019.
https://doi.org/10.1007/978-3-030-21548-4_8

there have been intensive researches in this direction with the appearance of improved schemes or ones with additional functionalities [9,12,16–18]. However, they are all proven to be secure in the random oracle model which does not exist in reality. Therefore it is necessary to construct such a scheme in the standard model.

Up to the present, there are only a few PKEET schemes in the standard model. Lee et al. [8] first proposed a generic construction of a PKEET scheme. Their method is to use a 2-level hierarchical identity-based encryption (HIBE) scheme together with a one-time signature scheme. The HIBE scheme is used for generating an encryption scheme and for equality test, and the signature scheme is used for making the scheme CCA2-secure, based on the method of transforming an identity-based encryption (IBE) scheme to a CCA2-secure encryption scheme of Canetti et al. [4]. As a result, they obtain a CCA2-secure PKEET scheme given that the underlying HIBE scheme is IND-sID-CPA secure and the one-time signature scheme is strongly unforgeable. From their generic construction, it is possible to obtain a PKEET in standard model under many hard assumptions via instantiations. In a very recent paper, Zhang et al. [22] proposed a direct construction of a CCA2-secure PKEET scheme based on pairings without employing strong cryptographic primitives such as HIBE schemes and strongly secure signatures as the generic construction of Lee et al. [8]. Their technique comes from a CCA2-secure public key encryption scheme by [7] which was directly constructed by an idea from IBE. A comparison with an instantiation from Lee et al. [8] on pairings shows that their direct construction is much more efficient than the instantiated one.

All aforementioned existing schemes base their security on the hardness of some number-theoretic assumptions which will be efficiently solved in the quantum era [14]. The generic construction by Lee et al. [8] is the first one with the possibility of yielding a post-quantum instantiation based on lattices, since lattice cryptography is the only one among other post-quantum areas up to present offers HIBE primitives, e.g., [1]. It is then still a question of either yielding an efficient instantiation or directly constructing a PKEET based on lattices.

Our Contribution: In this paper, we give a direct construction of a PKEET scheme based on lattices from IBE. According to the best of our knowledge, this is the first construction of a PKEET scheme based on lattices. We first employ the multi-bit full IBE by Agrawal et al. [1] and then directly transform it into a PKEET scheme. In our scheme, a ciphertext is of the form $\mathsf{CT} = (\mathsf{CT}_1, \mathsf{CT}_2, \mathsf{CT}_3, \mathsf{CT}_4)$ where $(\mathsf{CT}_1, \mathsf{CT}_3)$ is the encryption of the message $\mathbf{m}$, as in the original IBE scheme, and $(\mathsf{CT}_2, \mathsf{CT}_4)$ is the encryption of $H(\mathbf{m})$ in which H is a hash function. In order to utilize the IBE scheme, we employ a second hash function H' and create the *identity* $H'(\mathsf{CT}_1, \mathsf{CT}_2)$ before computing CT_3 and CT_4; see Sect. 3 for more details. Finally, we have proved that the proposed PKEET scheme is CCA2-secure. As compared to the previous constructions, the proposed one is computationally efficient due to the absence of exponentiation. But, the size of the public parameters is more.

2 Preliminaries

2.1 Public Key Encryption with Equality Test (PKEET)

In this section, we will recall the model of PKEET and its security model.

We remark that a PKEET system is a multi-user setting. Hence we assume that in our system throughout the paper, each user is assigned with an index i with $1 \leq i \leq N$ where N is the number of users in the system.

Definition 1 (PKEET)**.** *Public key encryption with equality test (PKEET) consists of the following polynomial-time algorithms:*

- $\mathsf{Setup}(\lambda)$*: On input a security parameter* λ *and set of parameters, it outputs the a pair of a user's public key* PK *and secret key* SK*.*
- $\mathsf{Enc}(\mathsf{PK}, \mathbf{m})$*: On input the public key* PK *and a message* $\mathbf{m}$*, it outputs a ciphertext* CT*.*
- $\mathsf{Dec}(\mathsf{SK}, \mathsf{CT})$*: On input the secret key* SK *and a ciphertext* CT*, it outputs a message* $\mathbf{m}'$ *or* $\perp$*.*
- $\mathsf{Td}(\mathsf{SK}_i)$*: On input the secret key* SK_i *for the user* U_i*, it outputs a trapdoor* td_i*.*
- $\mathsf{Test}(\mathsf{td}_i, \mathsf{td}_j, \mathsf{CT}_i, \mathsf{CT}_j)$*: On input two trapdoors* $\mathsf{td}_i, \mathsf{td}_j$ *and two ciphertexts* $\mathsf{CT}_i, \mathsf{CT}_j$ *for users* U_i *and* U_j *respectively, it outputs* 1 *or* 0*.*

Correctness. We say that a PKEET scheme is *correct* if the following three condition hold:

(1) For any security parameter λ, any user U_i and any message $\mathbf{m}$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{SK}_i, \mathsf{CT}_i) = \mathbf{m} \;\middle|\; \begin{array}{c}(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Setup}(\lambda) \\ \mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{PK}_i, \mathbf{m})\end{array}\right] = 1.$$

(2) For any security parameter λ, any users U_i, U_j and any messages $\mathbf{m}_i, \mathbf{m}_j$, it holds that:

$$\Pr\left[\mathsf{Test}\begin{pmatrix}\mathsf{td}_i \\ \mathsf{td}_j \\ \mathsf{CT}_i \\ \mathsf{CT}_j\end{pmatrix} = 1 \;\middle|\; \begin{array}{l}(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Setup}(\lambda) \\ \mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{PK}_i, \mathbf{m}_i) \\ \mathsf{td}_i \leftarrow \mathsf{Td}(\mathsf{SK}_i) \\ (\mathsf{PK}_j, \mathsf{SK}_j) \leftarrow \mathsf{Setup}(\lambda) \\ \mathsf{CT}_j \leftarrow \mathsf{Enc}(\mathsf{PK}_j, \mathbf{m}_j) \\ \mathsf{td}_j \leftarrow \mathsf{Td}(\mathsf{SK}_j)\end{array}\right] = 1$$

if $\mathbf{m}_i = \mathbf{m}_j$ regardless of whether $i = j$.

(3) For any security parameter λ, any users U_i, U_j and any messages $\mathbf{m}_i, \mathbf{m}_j$, it holds that

$$\Pr\left[\mathsf{Test}\begin{pmatrix}\mathsf{td}_i \\ \mathsf{td}_j \\ \mathsf{CT}_i \\ \mathsf{CT}_j\end{pmatrix} = 1 \;\middle|\; \begin{array}{l}(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Setup}(\lambda) \\ \mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{PK}_i, \mathbf{m}_i) \\ \mathsf{td}_i \leftarrow \mathsf{Td}(\mathsf{SK}_i) \\ (\mathsf{PK}_j, \mathsf{SK}_j) \leftarrow \mathsf{Setup}(\lambda) \\ \mathsf{CT}_j \leftarrow \mathsf{Enc}(\mathsf{PK}_j, \mathbf{m}_j) \\ \mathsf{td}_j \leftarrow \mathsf{Td}(\mathsf{SK}_j)\end{array}\right]$$

is negligible in λ for any ciphertexts CT_i, CT_j such that $\mathsf{Dec}(\mathsf{SK}_i, \mathsf{CT}_i) \neq \mathsf{Dec}(\mathsf{SK}_j, \mathsf{CT}_j)$ regardless of whether $i = j$.

Security Model of PKEET. For the security model of PKEET, we consider two types of adversaries:

- Type-I adversary: for this type, the adversary can request to issue a trapdoor for the target user and thus can perform equality tests on the challenge ciphertext. The aim of this type of adversaries is to reveal the message in the challenge ciphertext.
- Type-II adversary: for this type, the adversary cannot request to issue a trapdoor for the target user and thus cannot perform equality tests on the challenge ciphertext. The aim of this type of adversaries is to distinguish which message is in the challenge ciphertext between two candidates.

The security model of a PKEET scheme against two types of adversaries above is described in the following.

OW-CCA2 Security Against Type-I Adversaries. We illustrate the game between a challenger $\mathcal{C}$ and a Type-I adversary $\mathcal{A}$ who can have a trapdoor for all ciphertexts of the target user, say U_θ, that he wants to attack, as follows:

1. **Setup:** The challenger $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate the key pairs $(\mathsf{PK}_i, \mathsf{SK}_i)$ for all users with $i = 1, \cdots, N$, and gives $\{\mathsf{PK}_i\}_{i=1}^N$ to $\mathcal{A}$.
2. **Phase 1:** The adversary $\mathcal{A}$ may make queries polynomially many times adaptively and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{SK}}$: an oracle that on input an index i (different from θ), returns the U_i's secret key SK_i.
   - $\mathcal{O}^{\mathsf{Dec}}$: an oracle that on input a pair of an index i and a ciphertext CT_i, returns the output of $\mathsf{Dec}(\mathsf{SK}_i, \mathsf{CT}_i)$ using the secret key of the user U_i.
   - $\mathcal{O}^{\mathsf{Td}}$: an oracle that on input an index i, return td_i by running $\mathsf{td}_i \leftarrow \mathsf{Td}(\mathsf{SK}_i)$ using the secret key SK_i of the user U_i.
3. **Challenge:** $\mathcal{C}$ chooses a random message $\mathbf{m}$ in the message space and run $\mathsf{CT}_\theta^* \leftarrow \mathsf{Enc}(\mathsf{PK}_\theta, \mathbf{m})$, and sends CT_θ^* to $\mathcal{A}$.
4. **Phase 2:** $\mathcal{A}$ can query as in Phase 1 with the following constraints:
   - The index θ cannot be queried to the key generation oracle $\mathcal{O}^{\mathsf{SK}}$;
   - The pair of the index θ and the ciphertext CT_θ^* cannot be queried to the decryption oracle $\mathcal{O}^{\mathsf{Dec}}$.
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

$$\mathsf{Adv}^{\mathsf{OW\text{-}CCA2}}_{\mathcal{A},\mathrm{PKEET}}(\lambda) := \Pr[\mathbf{m} = \mathbf{m}'].$$

Remark 2. *If the message space is polynomial in the security parameter or the min-entropy of the message distribution is much lower than the security parameter then a Type-I adversary $\mathcal{A}$ with a trapdoor for the challenge ciphertext can reveal the message in polynomial-time or small exponential time in the security parameter, by performing the equality tests with the challenge ciphertext and all other ciphertexts of all messages generated by himself. Hence to prevent this attack, we assume that the size of the message space $\mathcal{M}$ is exponential in the security parameter and the min-entropy of the message distribution is sufficiently higher than the security parameter.*

IND-CCA2 Security Against Type-II Adversaries. We present the game between a challenger $\mathcal{C}$ and a Type-II adversary $\mathcal{A}$ who cannot have a trapdoor for all ciphertexts of the target user U_θ as follows:

1. **Setup:** The challenger $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate the key pairs $(\mathsf{PK}_i, \mathsf{SK}_i)$ for all users with $i = 1, \cdots, N$, and gives $\{\mathsf{PK}_i\}_{i=1}^N$ to $\mathcal{A}$.
2. **Phase 1:** The adversary $\mathcal{A}$ may make queries polynomially many times adaptively and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{SK}}$: an oracle that on input an index i (different from t), returns the U_i's secret key SK_i.
   - $\mathcal{O}^{\mathsf{Dec}}$: an oracle that on input a pair of an index i and a ciphertext CT_i, returns the output of $\mathsf{Dec}(\mathsf{SK}_i, \mathsf{CT}_i)$ using the secret key of the user U_i.
   - $\mathcal{O}^{\mathsf{Td}}$: an oracle that on input an index i (different from t), return td_i by running $\mathsf{td}_i \leftarrow \mathsf{Td}(\mathsf{SK}_i)$ using the secret key SK_i of the user U_i.
3. **Challenge:** $\mathcal{A}$ chooses 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 $\mathsf{CT}^*_{\theta,b} \leftarrow \mathsf{Enc}(\mathsf{PK}_\theta, \mathbf{m}_b)$ and sends $\mathsf{CT}^*_{\theta,b}$ to $\mathcal{A}$.
4. **Phase 2:** $\mathcal{A}$ can query as in Phase 1 with the following constraints:
   - The index t cannot be queried to the key generation oracle $\mathcal{O}^{\mathsf{SK}}$ and the trapdoor generation oracle $\mathcal{O}^{\mathsf{Td}}$;
   - The pair of the index θ and the ciphertext $\mathsf{CT}^*_{\theta,b}$ cannot be queried to the decryption oracle $\mathcal{O}^{\mathsf{Dec}}$.
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

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathcal{A},\mathrm{PKEET}} := \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

2.2 Lattices

Throughout the paper, we will mainly focus on integer lattices, which are 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

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

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 $\mathbf{u} \in \mathbb{Z}_q^n$

$$\begin{aligned}
\Lambda_q(A) &:= \left\{\mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \exists \mathbf{s} \in \mathbb{Z}_{\mathrm{q}}^{\mathrm{n}} \text{ where } \mathrm{A}^{\mathrm{T}}\mathbf{s} = \mathbf{e} \mod \mathrm{q}\right\} \\
\Lambda_q^{\perp}(A) &:= \{\mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathrm{A}\mathbf{e} = 0 \mod \mathrm{q}\} \\
\Lambda_q^{\mathbf{u}}(A) &:= \{\mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathrm{A}\mathbf{e} = \mathbf{u} \mod \mathrm{q}\}
\end{aligned}$$

Note that if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(A)$ then $\Lambda_q^{\mathbf{u}}(A) = \Lambda_q^{\perp}(A) + \mathbf{t}$.

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*

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

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:*

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

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

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

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 [11, 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 a matrix 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 matrix* $\mathbf{e} \in \mathbb{Z}^{(m+m_1)\times t}$ *distributed statistically close to* $\mathcal{D}_{\Lambda_q^U(F_1),\sigma}$ *where* $F_1 := (A \mid M)$. *In particular* $\mathbf{e} \in \Lambda_q^U(F_1)$, *i.e.,* $F_1 \cdot \mathbf{e} = U \mod q$.
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 matrix* $\mathbf{e} \in \mathbb{Z}^{(m+k)\times t}$ *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.,* $F_2 \cdot \mathbf{e} = U \mod q$.
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 [13].

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 [13] 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 Our PKEET Construction

3.1 Construction

Setup(λ): On input a security parameter λ, set the parameters q, n, m, σ, α as in Sect. 3.2

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 and $T_{A'}$ respectively.
2. Select $l + 1$ uniformly random $n \times m$ matrices $A_1, \cdots, A_l, B \in \mathbb{Z}_q^{n \times m}$.
3. Let $H : \{0,1\}^* \to \{0,1\}^t$ and $H' : \{0,1\}^* \to \{-1,1\}^l$ be hash functions.
4. Select a uniformly random matrix $U \in \mathbb{Z}_q^{n \times t}$.
5. Output the public key and the secret key

$$\mathsf{PK} = (A, A', A_1, \cdots, A_l, B, U), \quad \mathsf{SK} = (T_A, T_{A'}).$$

Encrypt$(\mathsf{PK}, \mathbf{m})$: On input the public key PK and a message $\mathbf{m} \in \{0,1\}^t$, do:

1. Choose a 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[1]

$$\mathbf{c}_1 = U^T \mathbf{s}_1 + \mathbf{x}_1 + \mathbf{m} \lfloor \frac{q}{2} \rfloor, \quad \mathbf{c}_2 = U^T \mathbf{s}_2 + \mathbf{x}_2 + H(\mathbf{m}) \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t.$$

3. Compute $\mathbf{b} = H'(\mathbf{c}_1 \| \mathbf{c}_2) \in \{-1,1\}^l$, and set

$$F_1 = (A | B + \sum_{i=1}^{l} b_i A_i), \quad F_2 = (A' | B + \sum_{i=1}^{l} b_i A_i).$$

4. Choose l uniformly random matrices $R_i \in \{-1,1\}^{m \times m}$ for $i = 1, \cdots, l$ and define $R = \sum_{i=1}^{l} b_i R_i \in \{-l, \cdots, l\}^{m \times m}$.
5. Choose $\mathbf{y}_1, \mathbf{y}_2 \in \overline{\Psi}_\alpha^m$ and set $\mathbf{z}_1 = R^T \mathbf{y}_1, \mathbf{z}_2 = R^T \mathbf{y}_2 \in \mathbb{Z}_q^m$.
6. Compute

$$\mathbf{c}_3 = F_1^T \mathbf{s}_1 + [\mathbf{y}_1^T | \mathbf{z}_1^T]^T, \mathbf{c}_4 = F_2^T \mathbf{s}_2 + [\mathbf{y}_2^T | \mathbf{z}_2^T]^T \in \mathbb{Z}_q^{2m}.$$

7. The ciphertext is

$$\mathsf{CT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4) \in \mathbb{Z}_q^{2t+4m}.$$

Decrypt$(\mathsf{PK}, \mathsf{SK}, \mathsf{CT})$: On input public key PK, private key SK and a ciphertext $\mathsf{CT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$, do:

1. Compute $\mathbf{b} = H'(\mathbf{c}_1 \| \mathbf{c}_2) \in \{-1,1\}^l$ and sample $\mathbf{e} \in \mathbb{Z}^{2m \times t}$ from

$$\mathbf{e} \leftarrow \mathsf{SampleLeft}(A, B + \sum_{i=1}^{l} b_i A_i, T_A, U, \sigma).$$

Note that $F_1 \cdot \mathbf{e} = U$ in $\mathbb{Z}_q^{n \times t}$.

[1] Note that for a message $\mathbf{m} \in \{0,1\}^t$, we choose a random binary string $\mathbf{m}'$ of fixed length t' large enough and by abusing of notation, we write $H(\mathbf{m})$ for $H(\mathbf{m}' \| \mathbf{m})$.

2. Compute $\mathbf{w} \leftarrow \mathbf{c}_1 - \mathbf{e}^T\mathbf{c}_3 \in \mathbb{Z}_q^t$.
3. For each $i = 1, \cdots, t$, compare w_i and $\lfloor \frac{q}{2} \rfloor$. If they are close, output $m_i = 1$ and otherwise output $m_i = 0$. We then obtain the message $\mathbf{m}$.
4. Sample $\mathbf{e}' \in \mathbb{Z}^{2m \times t}$ from

$$\mathbf{e}' \leftarrow \mathsf{SampleLeft}(A', B + \sum_{i=1}^{l} b_i A_i, T_{A'}, U, \sigma).$$

5. Compute $\mathbf{w}' \leftarrow \mathbf{c}_2 - (\mathbf{e}')^T\mathbf{c}_4 \in \mathbb{Z}_q^t$.
6. For each $i = 1, \cdots, t$, compare w_i' and $\lfloor \frac{q}{2} \rfloor$. If they are close, output $h_i = 1$ and otherwise output $h_i = 0$. We then obtain the vector $\mathbf{h}$.
7. If $\mathbf{h} = H(\mathbf{m})$ then output $\mathbf{m}$, otherwise output $\perp$.

Trapdoor(SK_i): On input a user U_i's secret key $\mathsf{SK}_i = (K_{i,1}, K_{i,2})$, it outputs a trapdoor $\mathsf{td}_i = K_{i,2}$.

Test$(\mathsf{td}_i, \mathsf{td}_j, \mathsf{CT}_i, \mathsf{CT}_j)$: On input trapdoors $\mathsf{td}_i, \mathsf{td}_j$ and ciphertexts $\mathsf{CT}_i, \mathsf{CT}_j$ for users U_i, U_j respectively, computes

1. For each i (resp. j), do the following:
   - Compute $\mathbf{b}_i = H'(\mathbf{c}_{i1} \| \mathbf{c}_{i2}) = (b_{i1}, \cdots, b_{il})$ and sample $\mathbf{e}_i \in \mathbb{Z}^{2m \times t}$ from

$$\mathbf{e_i} \leftarrow \mathsf{SampleLeft}(A_i', B_i + \sum_{k=1}^{l} b_{ik} A_{ik}, T_{A_i'}, U_i, \sigma).$$

     Note that $F_{i2} \cdot \mathbf{e}_i = U_i$ 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 each coordinate w_{ik} with $\lfloor \frac{q}{w} \rfloor$ and output $\mathbf{h}_{ik} = 1$ if they are close, and 0 otherwise. At the end, we obtain the vector $\mathbf{h}_i$ (resp. $\mathbf{h}_j$).
2. Output 1 if $\mathbf{h}_i = \mathbf{h}_j$ and 0 otherwise.

Theorem 4. *Our PKEET construction above is correct if H is a collision-resistant hash function.*

Proof. It is easy to see that if CT is a valid ciphertext of $\mathbf{m}$ then the decryption will always output $\mathbf{m}$. Moreover, if CT_i and CT_j are valid ciphertext of $\mathbf{m}$ and $\mathbf{m}'$ of user U_i and U_j respectively. Then the Test process checks whether $H(\mathbf{m}) = H(\mathbf{m}')$. If so then it outputs 1, meaning that $\mathbf{m} = \mathbf{m}'$, which is always correct with overwhelming probability since H is collision resistant. Hence our PKEET described above is correct. □

3.2 Parameters

We follow [1, Section 7.3] for choosing parameters for our scheme. Now for the system to work correctly we need to ensure

- the error term in decryption is less than $q/5$ with high probability, i.e., $q = \Omega(\sigma m^{3/2})$ and $\alpha < [\sigma l m \omega(\sqrt{\log m})]^{-1}$,
- that the $\mathsf{TrapGen}$ can operate, i.e., $m > 6n \log q$,

– that σ is large enough for SampleLeft and SampleRight, i.e., $\sigma > lm\omega(\sqrt{\log m})$,
– that Regev's reduction applies, 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).

Hence the following choice of parameters (q, m, σ, α) from [1] satisfies all of the above conditions, taking n to be the security parameter:

$$\begin{aligned} m &= 6n^{1+\delta}, \quad q = \max(2Q, m^{2.5}\omega(\sqrt{\log n})) \\ \sigma &= ml\omega(\sqrt{\log n}), \quad \alpha = [l^2 m^2 \omega(\sqrt{\log n})]^{-1} \end{aligned} \tag{1}$$

and round up m to the nearest larger integer and q to the nearest larger prime. Here we assume that δ is such that $n^\delta > \lceil \log q \rceil = O(\log n)$.

3.3 Security Analysis

In this section, we will prove that our proposed scheme is OW-CCA2 secure against Type-I adversaries (cf. Theorem 5) and IND-CCA2 secure against Type-II adversaries (cf. Theorem 6).

Theorem 5. *The PKEET with parameters $(q, n, m, \sigma, \alpha)$ as in (1) is OW-CCA2 secure provided that H is a one-way hash function, H' is a collision-resistant 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 OW-CCA2 game with advantage ϵ, then there is a probabilistic algorithm $\mathcal{B}$ that solves the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE problem with advantage ϵ' such that*

$$\epsilon' \geq \frac{1}{2q}\left(\epsilon - \frac{1}{2}\epsilon_{H',\mathsf{CR}} - \epsilon_{H,\mathsf{OW}}\right).$$

Here $\epsilon_{H',\mathsf{CR}}$ is the advantage of breaking the collision resistance of H' and $\epsilon_{H,\mathsf{OW}}$ is the advantage of breaking the one-wayness of H.

Proof. The proof is similar to that of [1, Theorem 25]. Assume that there is a Type-I adversary $\mathcal{A}$ who breaks the OW-CCA2 security of the PKKET scheme with non-negligible probability ϵ. We construct an algorithm $\mathcal{B}$ who solves the LWE problem using $\mathcal{A}$. Assume again that there are N users in our PKEET system. We now describe the behaviors of $\mathcal{B}$. Assume that θ is the target index of the adversary $\mathcal{A}$ and the challenge ciphertext is $\mathsf{CT}^*_\theta = (\mathsf{CT}^*_{\theta,1}, \mathsf{CT}^*_{\theta,2}, \mathsf{CT}^*_{\theta,3}, \mathsf{CT}^*_{\theta,4})$.

We will proceed the proof in a sequence of games. In game i, let W_i denote the event that the adversary $\mathcal{A}$ win the game. The adversary's advantage in Game i is $\Pr[W_i]$.

Game 0. This is the original OW-CCA2 game between the attacker $\mathcal{A}$ against the scheme and the OW-CCA2 challenger.

Game 1. This is similar to Game 0 except that in Phase 2 of Game 1, if the adversary queries the decryption oracel $\mathcal{O}^{\mathsf{Dec}}(\theta)$ of a ciphertext $\mathsf{CT}_\theta = (\mathsf{CT}_{\theta,1}, \mathsf{CT}_{\theta,2}, \mathsf{CT}_{\theta,3}, \mathsf{CT}_{\theta,4})$ such that $H'(\mathsf{CT}_{\theta,1} \| \mathsf{CT}_{\theta,2}) = \mathbf{b}^*$, where $\mathbf{b}^* = H'(\mathsf{CT}^*_{\theta,1} \| \mathsf{CT}^*_{\theta,2})$, but $\mathsf{CT}_\theta \neq \mathsf{CT}^*_\theta$ then the challenger aborts the game and returns a random guess. We denote this event by E_1. In this event, the adversary has found a collision for the hash function H' and so

$$\Pr[E_1] \leq \epsilon_{H',\mathsf{CR}}$$

where $\epsilon_{H'CR}$ is the advantage of the adversary $\mathcal{A}$ against the collision resistance of H'. Now the advantage of $\mathcal{A}$ in Game 1 is

$$\begin{aligned}
\Pr[W_1] &= \Pr[W_1|E_1]\Pr[E_1] + \Pr[W_1|\neg E_1]\Pr[\neg E_1] \\
&= \frac{1}{2}\Pr[E_1] + \Pr[W_0 \cap \neg E_1] \\
&= \frac{1}{2}\Pr[E_1] + \Pr[W_0] - \Pr[W_0 \cap E_1] \\
&\geq \Pr[W_0] - \frac{1}{2}\Pr[E_1] \\
&\geq \Pr[W_0] - \frac{1}{2}\epsilon_{H',\mathsf{CR}}
\end{aligned}$$

and hence

$$\Pr[W_0] - \Pr[W_1] \leq \frac{1}{2}\epsilon_{H',\mathsf{CR}}.$$

Game 2. This is similar to Game 1 except that at the challenge phase, $\mathcal{B}$ chooses two message $\mathbf{m}$ and $\mathbf{m}'$ in the message space and encrypt $\mathbf{m}$ in $\mathsf{CT}_{\theta,1}$ and $H(\mathbf{m}')$ in $\mathsf{CT}_{\theta,2}$. Other steps are similar to Game 1. Here we can not expect the behavior of $\mathcal{A}$. And since $\mathcal{A}$ has a trapdoor $T_{A'}$ and he can obtain $H(\mathbf{m}')$. At the end if $\mathcal{A}$ outputs $\mathbf{m}'$, call this event E_2, then $\mathcal{A}$ has broken the one-wayness of the hash function H. Thus

$$\Pr[E_2] \leq \epsilon_{H,\mathsf{OW}}$$

where $\epsilon_{H,\mathsf{OW}}$ is the advantage of $\mathcal{A}$ in breaking the one-wayness of H. Therefore we have

$$\begin{aligned}
\Pr[W_2] &= \Pr[W_2|E_2]\Pr[E_2] + \Pr[W_2|\neg E_2]\Pr[\neg E_2] \\
&= \Pr[W_2|E_2]\Pr[E_2] + \Pr[W_1]\Pr[\neg E_2] \\
&\geq \frac{1}{|\mathcal{M}|}\Pr[E_2] + \Pr[W_1] - \Pr[W_1]\Pr[E_2] \\
&\geq \Pr[W_1] - \Pr[E_2] \\
&\geq \Pr[W_1] - \epsilon_{H,\mathsf{OW}}
\end{aligned}$$

and hence

$$\Pr[W_1] - \Pr[W_2] \leq \epsilon_{H,\mathsf{OW}}.$$

Game 3. This is similar to Game 2 except the way the challenger $\mathcal{B}$ generates the public key for the user with index θ, as the following. Let $R_i^* \in \{-1,1\}^{m\times m}$ for $i = 1, \cdots, l$ be the ephemeral random matrices generated for the creation of the ciphertext CT_θ^*. In this game, the challenger chooses l matrices R_i^* uniformly random in $\{-1,1\}^{m\times m}$ and chooses l random scalars $h_i \in \mathbb{Z}_q$ for $i = 1, \cdots, l$. Then it generates A, A' and B as in Game 1 and constructs the matrices A_i for $i = 1, \cdots, l$ as

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

The remainder of the game is unchanged with R_i^*, $i = 1, \cdots, l$, used to generate the challenge ciphertext. Similar to the proof of [1, Theorem 25] we have that the A_i are close to uniform and hence they are random independent matrices in the view of the adversary as in Game 0. Therefore

$$\Pr[W_3] = \Pr[W_2].$$

Game 4. Game 4 is similar to Game 3 except that we add an abort that is independent of adversary's view. The challenger behaves as follows:

- The setup phase is identical to Game 3 except that the challenger also chooses random $h_i \in \mathbb{Z}_q$, $i = 1, \cdots, l$ and keeps it to itself.
- 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 $\mathsf{CT} = (\mathsf{CT}_1, \mathsf{CT}_2, \mathsf{CT}_3, \mathsf{CT}_4)$ to the decryption oracle $\mathcal{O}^{\mathsf{Dec}}$, the challenger checks whether $\mathbf{b} = H'(\mathsf{CT}_1 \| \mathsf{CT}_2)$ satisfies $1 + \sum_{i=1}^{h} b_i h_i \neq 0$ and $1 + \sum_{i=1}^{h} b_i^* h_i = 0$ where $\mathbf{b}^* = H'(\mathsf{CT}_{\theta,1}^* \| \mathsf{CT}_{\theta,2}^*)$. If not then the challenger overwrites $\mathbf{m}'$ with a fresh random message and aborts the game.
  2. **Artificial abort:** the challenger samples a message Γ such that $\Pr[\Gamma = 1]$ is calculated through a function $\mathcal{G}$ (defined as in [1]) evaluated through all the queries of $\mathcal{A}$. If $\Gamma = 1$ the challenger overwrites $\mathbf{m}'$ with a fresh random message and we say that the challenger aborted the game due to artificial abort; see [1] for more details.

A similar proof as in that of [1, Theorem 25] yields that

$$\Pr[W_4] \geq \frac{1}{2q}\Pr[W_3].$$

Game 5. We now change the way how A and B are generated in Game 4. In Game 5, A is a random matrix in $\mathbb{Z}_q^{n\times m}$ and B is generated through $\mathsf{TrapGen}(q, n)$ together with an associated trapdoor T_B for $\Lambda_q^{\perp}(B)$. The construction of A_i for $i = 1, \cdots, l$ remains the same as in Game 3, i.e., $A_i = AR_i^* - h_i B$. When $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{Dec}}(\theta, \mathsf{CT}_\theta)$ where $\mathsf{CT}_\theta = (\mathsf{CT}_{\theta,1}, \mathsf{CT}_{\theta,2}, \mathsf{CT}_{\theta,3}, \mathsf{CT}_{\theta,4})$, $\mathcal{B}$ performs as follows:

- $\mathcal{B}$ computes $\mathbf{b} = H'(\mathsf{CT}_{\theta,1} \| \mathsf{CT}_{\theta,2}) \in \{-1,1\}^l$ and set

$$F_\theta := (A|B + \sum_{i=1}^{l} A_i) = (A|AR + h_\theta B)$$

where

$$R \leftarrow \sum_{i=1}^{l} b_i R_i^* \in \mathbb{Z}_q^{n\times m} \quad \text{and} \quad h_\theta \leftarrow 1 + \sum_{i=1}^{l} b_i h_i \in \mathbb{Z}_q. \tag{2}$$

- If $h_\theta = 0$ then abort the game and pretend that the adversary outputs a random bit γ' as in Game 3.
- Set $\mathbf{e} \leftarrow \mathsf{SampleRight}(A, h_\theta B, R, T_B, U, \sigma) \in \mathbb{Z}_q^{2m\times t}$. Note that since h_θ is non-zero, and so T_B is also a trapdoor for $h_\theta B$. And hence the output $\mathbf{e}$ satisfies $F_\theta \cdot \mathbf{e} = U$ in $\mathbb{Z}_q^t$. Moreover, Theorem 2 shows that when $\sigma > \|\widetilde{T_B}\| s_R \omega(\sqrt{m})$ with $s_R := \|R\|$, the generated $\mathbf{e}$ is distributed close to $\mathcal{D}_{\Lambda_q^U}(F_\theta)$ as in Game 3.
- Compute $\mathbf{w} \leftarrow \mathsf{CT}_{\theta,1} - \mathbf{e}^T \mathsf{CT}_{\theta,3} \in \mathbb{Z}_q^t$. For each $i = 1, \cdots, t$, compare w_i with $\lfloor \frac{q}{2} \rfloor$, and output 1 if they are close, and output 0 otherwise. Then $\mathcal{B}$ can answer the decryption query $\mathcal{O}^{\mathsf{Dec}}(\theta, \mathsf{CT}_\theta)$ made by $\mathcal{A}$.

Game 5 is otherwise the same as Game 4. In particular, in the challenge phase, the challenger checks if b^* satisfies $1 + \sum_{i=1}^{l} b_i h_i = 0$. If not, the challenger aborts the game as in Game 4. Similarly, in Game 5, the challenger also implements an artificial abort in the guess phase. Since Game 4 and Game 5 are identical in the adversary's view, we have that

$$\Pr[W_5] = \Pr[W_4].$$

Game 6. Game 6 is identical to Game 5, except that the challenge ciphertext is always chosen randomly. And thus the advantage of $\mathcal{A}$ is always 0.

We now show that Game 5 and Game 6 are computationally indistinguishable. If the abort event happens then the games are clearly indistinguishable. We, therefore, consider only the queries that do not cause an abort.

Suppose now $\mathcal{A}$ has a non-negligible advantage in distinguishing Game 5 and Game 6. We use $\mathcal{A}$ to construct $\mathcal{B}$ to solve the LWE problem as follows.

Setup. First of all, $\mathcal{B}$ requests from $\mathcal{O}$ and receives, for each $j = 1, \cdots, t$ a fresh pair $(\mathbf{a}_i, d_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ and for each $i = 1, \cdots, m$, a fresh pair $(\mathbf{u}_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. $\mathcal{A}$ announces an index θ for the target user. $\mathcal{B}$ executes $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Setup}(\lambda)$ for $1 \le i \ne \theta \le N$. Then $\mathcal{B}$ constructs the public key for user of index θ 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 $\mathbf{u}_i$ for all $i = 1, \cdots, m$.
2. Assemble the first t unused the samples $\mathbf{a}_1, \cdots, \mathbf{a}_t$ to become a public random matrix $U \in \mathbb{Z}_q^{n\times t}$.
3. Run $\mathsf{TrapGen}(q, \sigma)$ to generate uniformly random matrices $A', B \in \mathbb{Z}_q^{n\times m}$ together with their trapdoor $T_{A'}$ and T_B respectively.
4. Choose l random matrices $R_i^* \in \{-1, 1\}^{m\times m}$ for $i = 1, \cdots, l$ and l random scalars $h_i \in \mathbb{Z}_q$ for $i = 1, \cdots, l$. Next it constructs the matrices A_i for $i = 1, \cdots, l$ as

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

Note that it follows from the leftover hash lemma [15, Theorem 8.38] that $A_1, \cdots, A_l$ are statistically close to uniform.

5. Set $\mathsf{PK}_\theta := (A, A', A_1, \cdots, A_l, B, U)$ to $\mathcal{A}$.

Then $\mathcal{B}$ sends the public keys $\{\mathsf{PK}_i\}_{i=1}^N$ to the adversary $\mathcal{A}$.

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

Challenge. Now $\mathcal{B}$ chooses random messages $\mathbf{m}^*$ and computes the challenge ciphertext $\mathsf{CT}^*_\theta = (\mathsf{CT}^*_{\theta,1}, \mathsf{CT}^*_{\theta,2}, \mathsf{CT}^*_{\theta,3}, \mathsf{CT}^*_{\theta,4})$ as follows:

1. Assemble $d_1, \cdots, d_t, v_1, \cdots, v_m$ from the entries of the samples to form $\mathbf{d}^* = [d_1, \cdots, d_t]^T \in \mathbb{Z}_q^t$ and $\mathbf{v}^* = [v_1, \cdots, v_m]^T \in \mathbb{Z}_q^m$.
2. Set $\mathsf{CT}^*_{\theta,1} \leftarrow \mathbf{d}^* + \mathbf{m}^* \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$.
3. Choose a uniformly random $\mathbf{s}_2 \in \mathbb{Z}_q^n$ and $\mathbf{x}_2 \leftarrow \overline{\Psi}_\alpha^t$, compute

$$\mathsf{CT}^*_{\theta,2} \leftarrow U^T \mathbf{s}_2 + \mathbf{x}_2 + H(\mathbf{m}^*) \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t.$$

4. Compute $\mathbf{b}^* = H'(\mathsf{CT}^*_{\theta,1} \| \mathsf{CT}^*_{\theta,2}) \in \{-1, 1\}^l$ and $R^* := \sum_{i=1}^l b_i^* R_i^* \in \{-l, \cdots, l\}^{m \times m}$.
5. Set

$$\mathsf{CT}^*_{\theta,3} := \begin{bmatrix} \mathbf{v}^* \\ (R^*)^T \mathbf{v}^* \end{bmatrix} \in \mathbb{Z}_q^{2m}.$$

6. Choose $\mathbf{y}_2 \leftarrow \overline{\Psi}_\alpha^m$ and set

$$\mathsf{CT}^*_{\theta,4} := \begin{bmatrix} (A')^T \mathbf{s}_2 + \mathbf{y}_2 \\ (AR^*)^T \mathbf{s}_2 + (R^*)^T \mathbf{y}_2 \end{bmatrix} \in \mathbb{Z}_q^{2m}.$$

Then $\mathcal{B}$ sends $\mathsf{CT}^*_\theta = (\mathsf{CT}^*_{\theta,1}, \mathsf{CT}^*_{\theta,2}, \mathsf{CT}^*_{\theta,3}, \mathsf{CT}^*_{\theta,4})$ to $\mathcal{A}$.

Note that in case of no abort, one has $h_\theta = 0$ and so $F_\theta = (A | AR^*)$. When the LWE oracle is pseudorandom, i.e., $\mathcal{O} = \mathcal{O}_{\mathbf{s}}$ then $\mathbf{v}^* = A^T \mathbf{s} + \mathbf{y}$ for some random noise vector $\mathbf{y} \leftarrow \overline{\Psi}_\alpha^m$. Therefore $\mathsf{CT}^*_{\theta,3}$ in Step 5 satisfies:

$$\mathsf{CT}^*_{\theta,3} := \begin{bmatrix} A^T \mathbf{s} + \mathbf{y} \\ (AR^*)^T \mathbf{s} + (R^*)^T \mathbf{y} \end{bmatrix} = (F_\theta)^T \mathbf{s} + \begin{bmatrix} \mathbf{y} \\ (R^*)^T \mathbf{y} \end{bmatrix}.$$

Moreover, $\mathbf{d}^* = U^T \mathbf{s} + \mathbf{x}$ for some $\mathbf{x} \leftarrow \overline{\Psi}_\alpha^t$ and therefore

$$\mathsf{CT}^*_{\theta,1} = U^T \mathbf{s} + \mathbf{x} + \mathbf{m}^* \lfloor \frac{q}{2} \rfloor.$$

One can easily see that

$$\mathsf{CT}^*_{\theta,4} = [A' | AR^*]^T \mathbf{s}_2 + \begin{bmatrix} \mathbf{y}_2 (R^*)^T \mathbf{y}_2 \end{bmatrix}.$$

Therefore CT^*_θ is a valid ciphertext.

When $\mathcal{O} = \mathcal{O}_\$$ we have that $\mathbf{d}^*$ is uniform in $\mathbb{Z}_q^t$ and $\mathbf{v}^*$ is uniform in $\mathbb{Z}_q^m$. Then obviously $\mathsf{CT}^*_{\theta,1}$ is uniform. It follows also from the leftover hash lemma (cf. [15, Theorem 8.38]) that $\mathsf{CT}^*_{\theta,3}$ is also uniform.

Guess. After Phase 2, $\mathcal{A}$ guesses if it is interacting with a Game 5 or Game 6. The simulator also implements the artificial abort from Game 5 and Game 6 and output the final guess as the answer to the LWE problem.

We have seen above that when $\mathcal{O} = \mathcal{O}_{\mathbf{s}}$ then the adversary's view is as in Game 5. When $\mathcal{O} = \mathcal{O}_{\$}$ then the view of adversary is as in Game 6. Hence the advantage ϵ' of $\mathcal{B}$ in solving the LWE problem is the same as the advantage of $\mathcal{A}$ in distinguishing Game 5 and Game 6. Since $\Pr[W_6] = 0$, we have

$$\Pr[W_5] = \Pr[W_5] - \Pr[W_6] \leq \epsilon'.$$

Hence combining the above results, we obtain that

$$\epsilon = \Pr[W_0] \leq \frac{1}{2}\epsilon_{H',\mathsf{CR}} + \epsilon_{H,\mathsf{OW}} + 2q\epsilon'$$

which implies

$$\epsilon' \geq \frac{1}{2q}\left(\epsilon - \frac{1}{2}\epsilon_{H',\mathsf{CR}} - \epsilon_{H,\mathsf{OW}}\right)$$

as desired. □

Theorem 6. *The PKEET with parameters $(q, n, m, \sigma, \alpha)$ as in (1) is IND-CCA2 secure provided that H' is a collision-resistant 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-CCA2 game with advantage ϵ, then there is a probabilistic algorithm $\mathcal{B}$ that solves the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE problem with advantage ϵ' such that*

$$\epsilon' \geq \frac{1}{4q}\left(\epsilon - \frac{1}{2}\epsilon_{H',\mathsf{CR}}\right)$$

where $\epsilon_{H',\mathsf{CR}}$ is the advantage of $\mathcal{A}$ in breaking the collision resistance of H'.

Proof. The proof is similar to that of Theorem 5. Assume that there is a Type-II adversary $\mathcal{A}$ who breaks the IND-CCA2 security of the PKKET scheme with non-negligible probability ϵ. We construct an algorithm $\mathcal{B}$ who solves the LWE problem using $\mathcal{A}$. Assume again that there are N users in our PKEET system. We now describe the behavior of $\mathcal{B}$. Assume that θ is the target index of the adversary $\mathcal{A}$ and the challenge ciphertext is $\mathsf{CT}^*_\theta = (\mathsf{CT}^*_{\theta,1}, \mathsf{CT}^*_{\theta,2}, \mathsf{CT}^*_{\theta,3}, \mathsf{CT}^*_{\theta,4})$.

We will proceed the proof in a sequence of games. In game i, let W_i denote the event that the adversary $\mathcal{A}$ correctly guesses the challenge bit. The adversary's advantage in Game i is $\left|\Pr[W_i] - \frac{1}{2}\right|$.

Game 0. This is the original IND-CCA2 game between the attacker $\mathcal{A}$ against the scheme and the IND-CCA2 challenger.

Game 1. This is similar to Game 1 in the proof of Theorem 5. Thus the advantage of $\mathcal{A}$ in Game 1 is

$$\left|\Pr[W_0] - \frac{1}{2}\right| - \left|\Pr[W_1] - \frac{1}{2}\right| \leq \frac{1}{2}\epsilon_{H',\mathsf{CR}}.$$

Game 2. This is similar to Game 3 in the proof of Theorem 5 and we have

$$\Pr[W_2] = \Pr[W_1].$$

Game 3. Game 3 is similar to Game 2 except that we add an abort as in the proof of Theorem 5. It follows from the proof of [1, Theorem 25] that

$$\left|\Pr[W_3] - \frac{1}{2}\right| \geq \frac{1}{4q}\left|\Pr[W_2] - \frac{1}{2}\right|.$$

Game 4. This game is similar to Game 5 in the proof of Theorem 5, and we have

$$\Pr[W_3] = \Pr[W_4].$$

Game 5. Game 5 is identical to Game 4, except that the challenge ciphertext is always chosen randomly. And thus the advantage of $\mathcal{A}$ is always 0.

We now show that Game 4 and Game 5 are computationally indistinguishable. If the abort event happens then the games are clearly indistinguishable. We, therefore, consider only the queries that do not cause an abort.

Suppose now $\mathcal{A}$ has a non-negligible advantage in distinguishing Game 4 and Game 5. We use $\mathcal{A}$ to construct $\mathcal{B}$ to solve the LWE problem similar to the proof of Theorem 5. Note that in the $\mathsf{IND\text{-}CCA2}$ game, we allow the adversary to query the trapdoor oracle $\mathcal{O}^{\mathsf{Td}}$. And since we generate A' together with $T_{A'}$ from $\mathsf{TrapGen}(q, n)$ and we can answer $T_{A'}$ to such queries.

We have seen above that when $\mathcal{O} = \mathcal{O}_{\mathbf{s}}$ then the adversary's view is as in Game 4. When $\mathcal{O} = \mathcal{O}_{\$}$ then the view of the adversary is as in Game 5. Hence the advantage ϵ' of $\mathcal{B}$ in solving the LWE problem is the same as the advantage of $\mathcal{A}$ in distinguishing Game 4 and Game 5. Since $\Pr[W_5] = \frac{1}{2}$, we have

$$\left|\Pr[W_4] - \frac{1}{2}\right| = |\Pr[W_4] - \Pr[W_5]| \leq \epsilon'.$$

Hence combining the above results, we obtain that

$$\epsilon = \left|\Pr[W_0] - \frac{1}{2}\right| \leq \frac{1}{2}\epsilon_{H',\mathsf{CR}} + 4q\epsilon'$$

which implies

$$\epsilon' \geq \frac{1}{4q}\left(\epsilon - \frac{1}{2}\epsilon_{H',\mathsf{CR}}\right)$$

as desired. □

4 Conclusion

In this paper, we propose a direct construction of PKEET based on the hardness of Learning With Errors problem. Efficiency is the reason to avoid the instantiation of lattice-based PKEET from the generic construction by Lee et

al. [8]. A concrete instantiation from [8] and comparative study are left for the complete version. In addition, our PKEET scheme can be further improved by utilizing improved IBE schemes [19,20] together with the efficient trapdoor generation [10] and faster Gaussian sampling technique [6], which we leave as future work.

Acknowledgement. The authors acknowledge the useful comments and suggestions of the referees. The first author would like to thank Hyung Tae Lee for sending him a copy of [22] and useful discussions, and acknowledges the support of the Start-Up Grant from University of Wollongong.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28
2. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_1
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, Proceedings, 26–28 February 2009, Freiburg, Germany, pp. 75–86 (2009)
4. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_13
5. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
6. Genise, N., Micciancio, D.: Faster Gaussian sampling for trapdoor lattices with arbitrary modulus. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 174–203. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_7
7. Lai, J., Deng, R.H., Liu, S., Kou, W.: Efficient CCA-secure PKE from identity-based techniques. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 132–147. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11925-5_10
8. Lee, H.T., Ling, S., Seo, J.H., Wang, H., Youn, T.Y.: Public key encryption with equality test in the standard model. Cryptology ePrint Archive, Report 2016/1182 (2016)
9. Lee, H.T., Ling, S., Seo, J.H., Wang, H.: Semi-generic construction of public key encryption and identity-based encryption with equality test. Inf. Sci. **373**, 419–440 (2016)
10. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
11. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In 45th Symposium on Foundations of Computer Science (FOCS 2004), Proceedings, 17–19 October 2004, Rome, Italy, pp. 372–381 (2004)

12. Ma, S., Zhang, M., Huang, Q., Yang, B.: Public key encryption with delegated equality test in a multi-user setting. Comput. J. **58**(4), 986–1002 (2015)
13. Regev, O.: 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, 22–24 May 2005, pp. 84–93 (2005)
14. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
15. Shoup, V.: A Computational Introduction to Number Theory and Algebra, 2nd edn. Cambridge University Press, Cambridge (2008)
16. Tang, Q.: Towards public key encryption scheme supporting equality test with fine-grained authorization. In: Parampalli, U., Hawkes, P. (eds.) ACISP 2011. LNCS, vol. 6812, pp. 389–406. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22497-3_25
17. Tang, Q.: Public key encryption schemes supporting equality test with authorisation of different granularity. IJACT **2**(4), 304–321 (2012)
18. Tang, Q.: Public key encryption supporting plaintext equality test and user-specified authorization. Secur. Commun. Netw. **5**(12), 1351–1362 (2012)
19. Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 32–62. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_2
20. Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 161–193. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_6
21. Yang, G., Tan, C.H., Huang, Q., Wong, D.S.: Probabilistic public key encryption with equality test. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 119–131. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11925-5_9
22. Zhang, K., Chen, J., Lee, H.T., Qian, H., Wang, H.: Efficient public key encryption with equality test in the standard model. Theor. Comput. **755**, 65–80 (2019)