Lattice-based Public Key Encryption with Multi-Ciphertexts Equality Test in Cloud Computing

Giang Linh Duc Nguyen[2], Dung Hoang Duong[1], Huy Quoc Le[1], Willy Susilo[1], and Nhan Thanh Nguyen[3]

[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, qhl576@uowmail.edu.au
[2] Faculty of Science and Engineering, Macquarie University, Sydney, NSW, Australia
duclinhgiang.nguyen@students.mq.edu.au
[3] Faculty of Civil Engineering, Ho Chi Minh City University of Technology, Vietnam
nt.nhan@hutech.edu.vn

Abstract. Nowadays, together with stormy technology advancement, billions of interconnected devices are constantly collecting data around us. In that fashion, privacy protection has become a major concern. The data must be in encrypted form before being stored on the cloud servers. As a result, the cloud servers are unable to perform calculations on encrypted data, such as searching and matching keywords. In the PKE-MET setting, a cloud server can perform an equality test on a number of ciphertexts which encrypted with the same designated number. In this paper, we propose, for the first time, an efficient construction of a quantum-safe PKE-MET system based on the hardness of the Learning with Errors (LWE) problem in the lattice setting. Furthermore, we also discuss the first lattice-base public key encryption with flexible multi-ciphertext equality test (PKE-FMET) constructions, which allow performing equality test on multiple ciphertexts whose designated numbers are less than a threshold number. Our proposed schemes are proven to be secure in the standard model.

Key words: Cloud computing, Multi-ciphertext quality test, Public key encryption, Lattice-based cryptography, Learning with Errors.

1 Introduction

The dramatic developments in technologies such as Cloud Infrastructures, the Internet of Things, and Big Data with millions of personal devices, have enormously impacted various business sectors and our daily lives. However, these personal devices do not have enough infrastructure and hardware power to execute intensive computations on collected data. In such a fashion, cloud servers

can store our collected data and perform intensive computations thanks to their extensible storage and computational powers.

Since data collected by personal devices are often very sensitive in terms of privacy, they should be encrypted before sending to the cloud servers. Consequently, the this step causes cloud servers unable to read through encrypted data and make effective computations. The idea of performing effective calculations directly on encrypted data without the need for decryption to preserve users' privacy has attracted a lot of attention from the research community. Many impressive cryptographic primitives / techniques have been proposed, such as searchable encryption [9], fully homomorphic encryption [5] and equality test [25], just to name a few, to address the aforementioned problem. Particularly, Yang et al. [25] introduced the definition of public-key encryption with equality test (PKEET), in which the equality of underlying message of two ciphertexts can be tested without the need of decrypting them. The functionality of PKEET has a wide range of applications, especially in smart city applications [24], cloud computing, and smart health care, such as the partition of encrypted emails [15], malware detection, and verifiability of encrypted data [3], and in wireless body area networks [19]. Another more straightforward example of the PKEET's applications is that people can find their friends who have the same interests without revealing them by matching their encrypted data with others. Subsequent efforts for PKEET have been devoted to satisfying different privacy requirements, improving efficiency, and extending/applying PKEET to other primitives.

Unfortunately, some scenarios in which the existing PKEET schemes are not practical due to their ineffectiveness and privacy disclosure. We can take the case of three users A, B, and C, in a group who want to check whether their ciphertexts are encrypted with the same message or not, for example. In this case, the traditional PKEET needs to perform precisely two equality tests. In addition, if the underlying messages are not the same, then the server can get unnecessary information (e.g., the server can know that users A and B have the same message while users B and C have different ones). In general, the computation cost linearly increases with the number of users. We will present such a formal problem in Section 1.2.

1.1 Related Works

Ever since Yang et al. [25] has introduced the first notion of PKEET, there have been a lot of works involving this oriented research to enhance privacy protection. The researchers add various restrictions on which a party can perform the equality test and can choose which type of ciphertexts can be performed equality test on [15]. Later, Ma et al. [16] improve the extension of authorization such that only the specified proxy which the user authorizes can perform the equality test on the user's ciphertexts. They introduce four types of authorization policies which are usually called the flexible authorization (FA) mechanism.

Many other researchers focused on improving the security and efficiency of PKEET. Zhang et al. [26] enhanced the efficiency of the PKEET scheme. It

achieves a shorter ciphertext size and trapdoor size and reduces the computation cost. The security is proven in the standard model (SDM) under the decisional bilinear Diffie–Hellman (DBDH) assumption. Lee et al. [14] introduced a generic construction for PKEET, which is secure in the standard model (SDM) and can be instantiated in the lattice setting. Their construction is based on a two-level hierarchical identity-based encryption (HIBE), a strong unforgeable one-time signature, and a cryptographic hash function. Duong et al. [10] have introduced an efficient direct construction PKEET based on lattices that are secure in SDM. Their method exploits the adaptive identity-based encryption (Full-IBE) scheme, which is a post-quantum instantiation based on lattices proposed by Agrawal et al. [1]. In the IBE setting, there also have been some works on equality tests, such as [11], [18] in which the latter also supports the FA mechanism. The equality test mechanism was also applied to other primitives such as signcryption [13], or certificate-less public key encryption [12]. Note that all the impressive works mentioned above focus on the equality test of two ciphertexts.

To enable an efficient and secure equality test among multiple ciphertexts, recently, Susilo et al. [23] have proposed a novel concept of public-key encryption with a multiple-ciphertext equality test (PKE-MET) to avoid the aforementioned drawbacks of the traditional PKEET. In PKE-MET, each ciphertext has a designated number, say β, such that the equality test can only be performed on β ciphertexts, including this ciphertext itself. Furthermore, all the ciphertexts must have the same designated number β. In PKE-MET, the honest but curious sever performs the equality test on multiple ciphertexts at once and extracts nothing but whether the underlying messages are equal or not. They call this *anti-disclose information* (AntiD) property. To make the PKE-MET more flexible and practical, an extended version called Public Key Encryption with Flexible Multiple Equality Tests (PKE-FMET) was introduced. The PKE-FMET scheme allows performing equality test on γ ciphertexts whose designated number $\beta \leq \gamma$. It also archives the AntiD property. The schemes are proved to be secure in ROM under the Diffie-Hellman assumption.

In addition, in 1994, Petter Shor [21] introduced a breakthrough result demonstrating that quantum algorithms could easily solve number-theoretic assumptions such as factoring and discrete logarithm problems. This means that, the classical cryptosystems based on these number-theoretic assumptions is not secure any more in the post-quantum era. Therefore, we should start preparing for the transition to post-quantum cryptography now.

We summarize some related works in Table 1. To the best of our knowledge, there has been no post-quantum secure PKE-MET and PKE-FMET schemes yet. Therefore, it is necessary to construct PKE-MET and PKE-FMET in the standard model and still secure even in the upcoming quantum era. In this paper, we attempt to use lattice problems as a vital ingredient for proposing PKE-MET and PKE-FMET constructions.

Table 1. A comparison of some encryption scheme with equality test.

Literature	Assumption	Model	AntiD	FMET
Ma et al. [16]	CDH	ROM	×	×
Zhang et al. [26]	DBDH	SDM	×	×
Susilo et al. [23]	CDH	ROM	✓	✓
Duong et al. [10]	LWE	SDM	×	×
This work	LWE	SDM	✓	✓

FMET stands for flexible multi-ciphertext equality test which will be discussed later.

1.2 PKE-(F)MET: A Use Case

In this section, we discuss a typical scenario in cloud computing and then propose one solution using PKE-(F)MET. Due to the COVID-19 pandemic, there is a need of monitoring community health in a city, region, or larger location. The IoT devices play an essential role by collecting the citizens' health metrics such as heartbeat rate, temperature, sleep cycle, and body hydration. Individuals in the community can send their personal health data and location history into a cloud server which a government organization can manage. To protect user's privacy, the data must be encrypted before sending it to the cloud server. The Centers for Disease Control and Prevention (CDC) can analyze citizen data to accurately detect the current trend of the virus and propose correct solutions to prevent the virus spreading. In addition, Other medical agencies can also perform analysis on the data for different categories and indicators to fasten their medicine production test. Furthermore, there is an option for the citizen to give their consent for other parties to use their data for analysis. They can opt in or opt out anytime, which gives the community more control of their past and/or data.

The scenario above is then transformed into a cloud computing model formally stated below.

Cloud Model The cloud model consists of the following parties (as shown in Figure 1).

- *End-user*: Any user can send encrypted data to the cloud server. They gives their consent for analyzing their data via sending their token to the server.
- *Cloud service*: A cloud infrastructure stores encrypted data. It then can perform various computations on the stored data, such as performing equality tests.
- *Third party*: A third party can be any public or private organization interested in computing the encrypted data on the cloud server.

Problem The Health Officer of the CDC would like to perform some statistical computations on the collected data stored in the cloud server. To do this, the

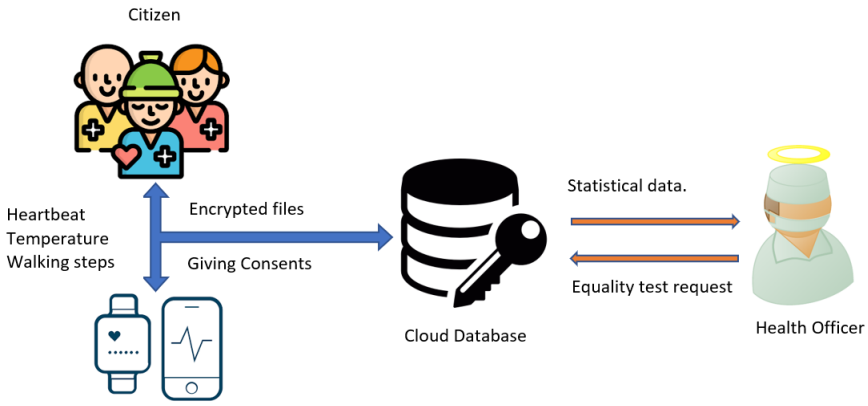

Fig. 1. Cloud Model

cloud server may be required to test the equality of underlying messages between several ciphertexts from the different citizens. The traditional PKEET technology can provide a mechanism for an authorized server to check the equality between pairs of ciphertexts per execution. This mechanism has two weaknesses: (i) computational inefficiency and (ii) disclosing the users' privacy.

For instance, provided that there are t users having messages $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_t$ with corresponding ciphertexts $\mathsf{CT}_1, \mathsf{CT}_2, \ldots, \mathsf{CT}_t$ respectively. The cloud sever can check if $\mathbf{m}_1 = \mathbf{m}_2 = \cdots = \mathbf{m}_t$ by testing the equality for every pair of ciphertexts $(\mathsf{CT}_1, \mathsf{CT}_2), (\mathsf{CT}_2, \mathsf{CT}_3), \ldots, (\mathsf{CT}_{t-1}, \mathsf{CT}_t)$ sequentially. It is clear that the mechanism is not effective. In addition, the cloud server can learn more information rather than the equality of all underlying messages. For example, suppose that $\mathbf{m}_1 = \mathbf{m}_2 = \cdots = \mathbf{m}_{t-1} \neq \mathbf{m}_t$, then the cloud sever knows extra information that the first $t-1$ users have the same messages while the last user does not.

Solution To mitigate the weaknesses above, the proposed the public key encryption with multiple ciphertexts equality test (PKE-MET) concept can be applied. In the PKE-MET system, the cloud service provider executes equality test requests on multiple ciphertexts to deduct nothing but only whether the underlying messages are equal or not. In this paper, we attempt to provide an effective construction of the PKE-MET system under the hardness of the LWE problem, which is believed to be secure against even large-scale quantum computers; see Section 3 for the detail.

The PKE-MET system is illustrated in Figure 2. For $i \in \{1, 2, \ldots, \gamma\}$, citizen i uploads a ciphertext CT_i with a designated number γ into the cloud database and then give their consent by providing token tk_i. The Health Officer can then make an equality test request to the cloud server for γ ciphertexts to get statistical information. The cloud server then executes the request and returns value 1 or 0 to the Health Officer, indicating that the underlying messages of these

ciphertexts are the same or not. Note that the cloud server can only perform the equality test on exactly γ ciphertexts $\mathsf{CT}_1, \mathsf{CT}_2, \ldots, \mathsf{CT}_\gamma$ whose designated number is γ.

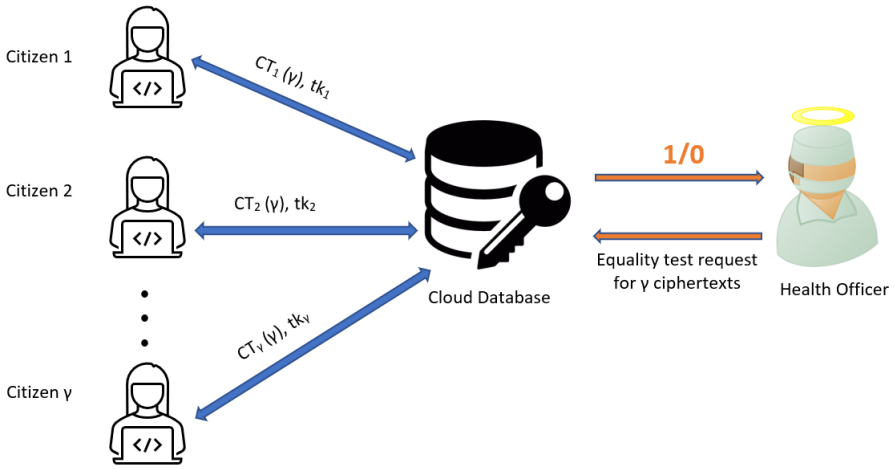

Fig. 2. PKE-MET system

The PKE-MET system can perform the equality test on multiple ciphertexts. However, it is not applicable for practical scenario where the ciphertexts can have different designated number. Therefore, the extended PKE with flexible MET (PKE-FMET) is proposed in [23]. The PKE-FMET system can be shown as Figure 3 where the authorized cloud sever can execute the equality test on γ ciphertexts $\mathsf{CT}_1, \mathsf{CT}_2, \ldots, \mathsf{CT}_\gamma$ whose designated numbers are $\beta_1, \beta_2, \ldots, \beta_\gamma$ respectively. The constraint is that $\beta \leq \gamma$ for $\beta = \max\{\beta_1, \beta_2, \ldots, \beta_\gamma\}$.

The proposed systems satisfy the following properties:

- *Correctness*: The proposed systems can correctly perform equality tests on ciphertexts that satisfy the requirements.
- *Security*: For ciphertexts that do not satisfy the requirement, the equality test will return $\perp$.
- *Efficiency*: The algorithm executes equality test can be performed efficiently in the context of multiple ciphertexts on cloud computing.

In this paper, we also extend our lattice-based PKE-MET construction to build PKE-FMET construction; see Section 4 for the detail.

For simplicity, we draw lines indicating the citizen gives their consents (tokens) to the cloud server in Figure 2 and Figure 3. However, a separate entity that handles token management is required for practical scenarios. The token management server will receive from the users and then distributes the token to a trusted person who then can calculate the statistical data by executing equality tests on ciphertexts. It also performs necessary actions when the citizens

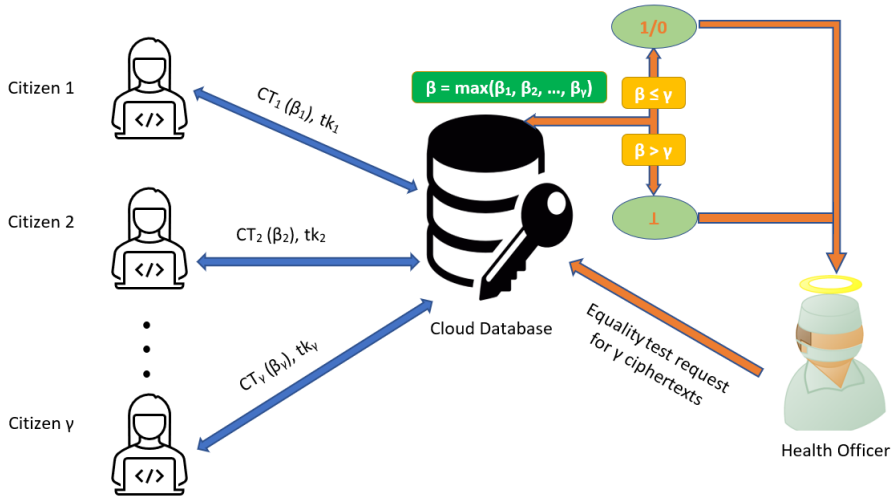

Fig. 3. PKE-FMET system

opt-in and opt-out from giving their consent. We leave the discussion on token management for future research.

1.3 Our Contribution and Technical Overview.

We propose the first concrete construction of a PKE-MET scheme secure in the standard model based on the hardness assumption of learning with errors problem in lattices. Our scheme is proved to be IND-CPA secure. However, one can modify it to achieve IND-CCA2 security using the BCHK's transformation [7].

Roughly speaking, to encrypt a message via the PKEET of [10], one samples a random identity vector $\mathbf{b} = (b_1, \cdots, b_\ell) \in \{-1, 1\}^\ell$ and then computes $F_1 := (A|B + \sum_{i=1}^{\ell} b_i A_i)$, which is then used together with the dual-Regev framework for encrypting the message. The vector $\mathbf{b}$ together with ℓ random scalars h_i will be chosen at the setup phase to support the security reduction from the learning with errors (LWE) problem to the IND-CPA and OW-CPA security; see Section 2 for the definitions and security models. They are also used to perform abort checks which are useful in our security analysis.

To support multi-ciphertext equality test, an one-way hash function takes a message $\mathbf{m}$ and designated number β to produce a number, say f_0. This number is then taken together with $\mathbf{m}$ and β as input for generating the next number, say f_1, and so on. After β iterations, the encryption algorithm creates a list of numbers $f_0, \cdots, f_{\beta-1}$. These numbers then play the role of coefficients of a predefined polynomial $f(x)$, i.e., $f(x) = f_0 + f_1 x + \cdots + f_{\beta-1} x^{\beta-1}$. We then compute and encrypt $f(\delta)$ and δ in the ciphertext $\mathbf{c}_2$ where δ is chosen randomly. The pair of $(f(\delta), \delta)$ is later utilized to reconstruct the Vandermonde matrix with variables $f_0, \cdots, f_{\beta-1}$. The solution to the Vandermonde matrix is the key to

Table 2. Comparison of our PKE-MET with other PKEET constructions to execute equality test on β ciphertexts.

Scheme	CT size	PK size	SK size	#Test
Lee's PKEET [14]	$8m + 2t + 2mt$	$(\ell+3)mn+nt$	$2m^2$	$\beta - 1$
Duong's PKEET [10]	$2t + 4m$	$(\ell+3)mn+nt$	$2m^2$	$\beta - 1$
Our PKE-MET	$2.5t+4m+\ell+\lambda$	$(\ell+3)mn+nt$	$2m^2$	1
Our PKE-FMET	$(\omega - \beta + 2)(t + 2m) + \ell + \lambda$	$(\ell+3)mn+nt$	$2m^2$	1

Data sizes are in the number of field elements. Here:
t is the length of messages.
ℓ is the length of identity vectors.
λ is the length of after image in cryptographic hash function.
β is the designated number for each ciphertext.
ω is the maximal number of ciphertexts can be used in equality test which only applicable in a PKE-FMET construction.
Test: Number of test round needed to execute equality test on β ciphertexts.

perform multi-ciphertext equality tests later. The key idea of this algorithm is inherited from Shamir (β, γ)-threshold secret sharing, where β is the designated number of the ciphertext, and γ is the required ciphertexts to perform multi-ciphertext equality test.

The ciphertext in our scheme has the form of $\mathsf{CT} = (\mathbf{b}, \beta, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5)$, where $(\mathbf{c}_1, \mathbf{c}_3)$ is the encryption of message $\mathbf{m}$, $(\mathbf{c}_2, \mathbf{c}_4)$ is the encryption of $(f(\delta), \delta)$, and $\mathbf{c}_5$ is the result of collision-resistant hash function we use to verify the equality test.

To make the multiple equality tests even more practical in cloud computing context, we proposed the Public Key Encryption with Flexible Multiple Equality Test (PKE-FMET) scheme. We contribute our ideas of extending PKE-MET to archive PKE-FMET in Section 4. That section also presents the strengths and weaknesses of our ideas.

To summary, we present in Table 1.3 a comparison of our PKE-MET with the lattice-based PKEETs of Lee et al. [14] and Duong et al. [10] in term of storage cost. As you can see, the our schemes take more data storage to archive the AntiD property and be able to execute equality test on multiple ciphertexts only once.

This paper is organized as follows. Section 2 describes the PKE-MET's definition and security model together with the basic integer lattice theory and some useful sampling algorithms. Section 3 contains our proposed lattice-based PKE-MET construction along with the notation list used throughout this paper. Then, in Section 4, we extend the PKE-MET scheme to PKE-FMET to support a flexible number of ciphertexts on which the authorized server can perform equality tests on. Finally, Section 5 presents our conclusion and future works.

2 Preliminaries

2.1 Public key encryption with multi-ciphertext equality test

In this section, we will recall the model of PKE-MET and its security model.

Definition 1. *An PKE-MET scheme consists of the following polynomial-time algorithms:*

- $\mathsf{Setup}(\lambda)$*: On input a security parameter* λ*, it outputs a system parameter* PP*.*
- $\mathsf{KeyGen}(pp)$*: On input* PP *parameter, it outputs a pair of public key and secret key* $(\mathsf{PK}, \mathsf{SK})$*.*
- $\mathsf{Enc}(\mathsf{PK}, \mathbf{m}, \beta)$*: On input public key* PK*, message* $\mathbf{m}$ *and a designated number* β*, it outputs a ciphertext* CT *of message* $\mathbf{m}$ *with the designated number* β *such that the equality test can only performed between the* CT *and* $\beta-1$ *other ciphertexts, whose designated number* β*.*
- $\mathsf{Dec}(\mathsf{SK}, \mathsf{CT})$*: On input ciphertext* CT*, and secret key* SK*, it outputs a message* $\mathbf{m}$ *or* $\perp$*.*
- $\mathsf{Aut}(\mathsf{SK})$*: On input secret key* SK*, it outputs a token* TK*, which will be used to authorize cloud server to perform equality test on the ciphertexts of the user who own the secret key* SK*.*
- $\mathsf{Test}(\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma)$*: On input* γ *ciphertexts* CT_i*, whose designated number* β_i *and* γ *token key* TK_i *for* $i \in \{1, \cdots, \gamma\}$*. It outputs* $\perp$ *if the equation* $\beta_1 = \cdots = \beta_\gamma$ *does not hold. Otherwise, it output 1 or 0, meaning the underlying messages* $\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma$ *are equal or not respectively.*

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

(1) For any security parameter λ, any message $\mathbf{m} \in \mathcal{M}$ and any number $\beta \in \mathbb{Z}_q$:

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

(2) For any security parameter λ, any message $\mathbf{m} \in \mathcal{M}$, any number $\gamma \in \mathbb{Z}_q$, and $i \in \{1, \cdots, \gamma\}$, it holds that:

$$\Pr\left[\mathsf{Test}\left(\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma\right) = 1 \,\middle|\, \begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(\lambda) \\ (\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ \mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{PK}_i, \mathbf{m}, \gamma) \\ \mathsf{TK}_i \leftarrow \mathsf{Aut}(\mathsf{SK}_i) \end{array}\right]$$

with overwhelming probability.

(3) For any security parameter λ, any message $\mathbf{m} \in \mathcal{M}$, any number $\gamma \in \mathbb{Z}_p$, and $i \in \{1, \cdots, \gamma\}$, it holds that:

$$\Pr\left[\mathsf{Test}\left(\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma\right) = 0 \;\middle|\; \begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(\lambda) \\ (\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ \mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{PK}, \mathbf{m}_i, \gamma) \\ \mathsf{TK}_i \leftarrow \mathsf{Aut}(\mathsf{SK}_i) \end{array}\right]$$

with overwhelming probability, where the equation $\mathbf{m}_1 = \cdots = \mathbf{m}_\gamma$ does not hold.

Security model of PKE-MET. For the PKE-MET security model, we consider three 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 user. Their goal is to reveal the plaintext 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 user. Their goal is to distinguish which message is in the challenge ciphertext between two candidates.
- Type-III adversary: the adversaries can perform authorization (equality) tests on the challenge ciphertext by requesting to obtain a trapdoor for authorization of the target users. Their goal is to perform the equality test on γ ciphertexts $\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma$, where all the designated numbers of these ciphertexts are β with $\beta > \gamma$.

OW-CPA security against Type-I adversaries. We present the game between a challenger $\mathcal{C}$ and a Type-I adversary $\mathcal{A}$ as follows

1. **Setup:** $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate the pair $(\mathsf{PK}_i, \mathsf{SK}_i)$ for all users with $i \in \{1, \cdots, N\}$, and sends the public keys set $\{\mathsf{PK}_i\}_{i=1}^N$ to $\mathcal{A}$.
2. **Phase 1:** $\mathcal{A}$ may adaptively make queries many times and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{SK}}$: on input an index i, $\mathcal{O}^{\mathsf{SK}}$ returns the target user $\mathcal{U}_i$'s secret key SK_i.
   - $\mathcal{O}^{\mathsf{Aut}}$: on input of an index i, $\mathcal{O}^{\mathsf{Aut}}$ returns $\mathsf{TK}_i = \mathsf{Aut}(\mathsf{SK}_i)$ where SK_i is the secret key of user $\mathcal{U}_i$.
3. **Challenge:** $\mathcal{A}$ selects a target user $\mathcal{U}_\theta$, which was never queried to the $\mathcal{O}^{\mathsf{SK}}$, and send to $\mathcal{C}$ who then chooses a random message $\mathbf{m} \in \mathcal{M}$ and a designated number β, then computes and sends $\mathsf{CT}^*_\theta \leftarrow \mathsf{Enc}(\mathsf{PK}^*_\theta, \mathbf{m}, \beta)$ to $\mathcal{A}$.
4. **Phase 2:** same as in Phase 1 except that $\mathcal{A}$ have following restrictions:
   - $\mathcal{A}$ cannot query to oracle $\mathcal{O}^{\mathsf{SK}}$ for the user $\mathcal{U}_\theta$.
5. **Guess:** $\mathcal{A}$ outputs $\mathbf{m}'$.

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

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

IND-CPA security against Type-II adversaries. We illustrate the game between a challenger $\mathcal{C}$ and a Type-II adversary $\mathcal{A}$ as follows

1. **Setup:** $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate the pair $(\mathsf{PK}_i, \mathsf{SK}_i)$ for all users with $i \in \{1, \cdots, N\}$, and sends the public keys set $\{\mathsf{PK}_i\}_{i=1}^{N}$ to $\mathcal{A}$.
2. **Phase 1:** $\mathcal{A}$ may adaptively make queries many times and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{SK}}$: on input an index i, $\mathcal{O}^{\mathsf{SK}}$ returns the user $\mathcal{U}_i$'s secret key SK_i.
   - $\mathcal{O}^{\mathsf{Aut}}$: on input an index i, $\mathcal{O}^{\mathsf{Aut}}$ returns $\mathsf{TK}_i = \mathsf{Aut}(\mathsf{SK}_i)$ where SK_i is the secret key of user $\mathcal{U}_i$.
3. **Challenge:** $\mathcal{A}$ selects a target user $\mathcal{U}_\theta$, which was never queried to the $\mathcal{O}^{\mathsf{SK}}$ and $\mathcal{O}^{\mathsf{Aut}}$ oracles in Phase 1, and two messages $\mathbf{m}_0$ $\mathbf{m}_1$ of same length and pass to $\mathcal{C}$. Challenger $\mathcal{C}$ then selects a random bit $b \in \{0, 1\}$, a designated number β, compute and sends $\mathsf{CT}_\theta^* \leftarrow \mathsf{Enc}(\mathsf{PK}_\theta^*, \mathbf{m}_b, \beta)$ to $\mathcal{A}$.
4. **Phase 2:** same as in Phase 1 except that $\mathcal{A}$ have following restrictions:
   - $\mathcal{A}$ cannot query to oracle $\mathcal{O}^{\mathsf{SK}}$ and $\mathcal{O}^{\mathsf{Aut}}$ for the user $\mathcal{U}_\theta$.
5. **Guess:** $\mathcal{A}$ outputs b'.

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

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

Number game security against Type-III adversaries. We illustrate the game between a challenger $\mathcal{C}$ and a Type-III adversary $\mathcal{A}$, who have a trapdoor for all ciphertexts of all users.

1. **Setup:** $\mathcal{C}$ runs $\mathsf{Setup}(\lambda)$ to generate the pair $(\mathsf{PK}_i, \mathsf{SK}_i)$ for all users with $i \in \{1, \cdots, N\}$, and sends the public keys set $\{\mathsf{PK}_i\}_{i=1}^{N}$ to $\mathcal{A}$.
2. **Phase 1:** $\mathcal{A}$ may adaptively make queries many times and in any order to the following oracles:
   - $\mathcal{O}^{\mathsf{SK}}$: on input an index i, $\mathcal{O}^{\mathsf{SK}}$ returns the user $\mathcal{U}_i$'s secret key SK_i.
   - $\mathcal{O}^{\mathsf{Aut}}$: on input of an index i, $\mathcal{O}^{\mathsf{Aut}}$ returns $\mathsf{TK}_i = \mathsf{Aut}(\mathsf{SK}_i)$ where SK_i is the secret key of user $\mathcal{U}_i$.
3. **Challenge:** $\mathcal{A}$ selects a list of γ users $\{\mathcal{U}_i\}_{i=1}^{\gamma}$, which was never queried to the $\mathcal{O}^{\mathsf{SK}}$ oracle in Phase 1 and a designated number β where $\gamma < \beta$ and pass to $\mathcal{C}$. Challenger $\mathcal{C}$ then chooses two messages $\mathbf{m}_0$, $\mathbf{m}_1$ of same length, and selects a random bit $b \in \{0, 1\}$, then compute

   $$(\mathsf{CT}_1^*, \cdots, \mathsf{CT}_i^*, \cdots, \mathsf{CT}_\gamma^*) \leftarrow (\mathsf{Enc}(\mathsf{PK}_1, \mathbf{m}_0, \beta), \cdots, \mathsf{Enc}(\mathsf{PK}_i, \mathbf{m}_b, \beta), \cdots, \mathsf{Enc}(\mathsf{PK}_\gamma, \mathbf{m}_0, \beta))$$

   where $i \in \{1, \cdots, \gamma\}$, and sends $(\mathsf{CT}_1^*, \cdots, \mathsf{CT}_i^*, \cdots, \mathsf{CT}_\gamma^*)$ to $\mathcal{A}$.
4. **Phase 2:** same as in Phase 1 except that $\mathcal{A}$ have following restrictions:
   - $\mathcal{A}$ cannot query to oracle $\mathcal{O}^{\mathsf{SK}}$ for the list of users $\{\mathcal{U}_i\}_{i=1}^{\gamma}$.
5. **Guess:** $\mathcal{A}$ outputs b'.

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

$$\mathsf{Adv}_{\mathcal{A},\text{PKE-MET}}^{\mathsf{Number}} := \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

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

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

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

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

Note that if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(A)$ then $\Lambda_q^{\mathbf{u}}(A) = \Lambda_q^{\perp}(A) + \mathbf{t}$. 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 [4] 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*

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

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:*

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

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 [17, Lemma 4.4] . The second one is from [8] 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 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., $F_1 \cdot \mathbf{e} = U \mod q$.*
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., $F_2 \cdot \mathbf{e} = U \mod q$. 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 [20].

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 [20] 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 Lattice-based PKE-MET Construction

In this section, we present a lattice-based PKE-MET construction. In the PKE-MET scheme, the authorized cloud server can execute the equality test on exactly β ciphertexts whose designated number is β.

3.1 Proposed construction

Setup(λ): Taking input a security parameter λ, the setup algorithm chooses parameters q, n, m, σ, α as in Section 3.5, where $\lceil \log_2 q \rceil = \tau$. Also, it chooses cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q$, $H_2 : \{0,1\}^* \to \{-1,1\}^\lambda$. Finally, it outputs the system parameters

$$pp = \{q, n, m, \sigma, \alpha, \tau, H_1, H_2\}.$$

KeyGen(pp): Taking input a system parameter pp, the setup algorithm generate a pair of public and private keys as follows:

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 $A_1, \cdots, A_\ell, B \in \mathbb{Z}_q^{n \times m}$.
3. Select a uniformly random matrix $U \in \mathbb{Z}_q^{n \times t}$.
4. Output the public key and the private key
   $\mathsf{PK} = (A, A', A_1, \cdots, A_\ell, B, U)$, $\mathsf{SK} = (T_A, T_{A'})$.

Encrypt$(\mathsf{PK}, \mathbf{m}, \beta)$: On input a public key PK, a message $\mathbf{m} \in \{0,1\}^t$ and a designated number $\beta \in \mathbb{Z}_q$, the encryption algorithm does:

1. Compute the following values in order

$$f_0 = H_1(\mathbf{m}||\mathsf{bin}(\beta)),\ f_1 = H_1(\mathbf{m}||\mathsf{bin}(\beta)||f_0),\ \cdots,$$

$$f_{\beta-1} = H_1(\mathbf{m}||\mathsf{bin}(\beta)||f_0||\cdots||f_{\beta-2}).$$

   with $\mathsf{bin}(\beta) \in \{0,1\}^\tau$ is the binary representation of β.
2. Consider the polynomial $f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{\beta-1} x^{\beta-1} \in \mathbb{Z}_q$, for $x \in \mathbb{Z}_q$.
3. Randomly choose $\delta \in \mathbb{Z}_q$ and compute $f(\delta) \in \mathbb{Z}_q$.

4. Choose uniformly random $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^n$ and $\mathbf{x}_1, \mathbf{x}_2 \leftarrow \overline{\Psi}_\alpha^t$, compute

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

$$\mathbf{c}_2 = U^T\mathbf{s}_2 + \mathbf{x}_2 + (\mathsf{bin}(\delta)||\mathsf{bin}(f(\delta)))\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q^t.$$

5. Choose randomly $\mathbf{b} = (b_1, \cdots, b_\ell) \in \{-1, 1\}^\ell$, set
$F_1 = (A|B + \sum_{i=1}^{\ell} b_i A_i)$,
$F_2 = (A'|B + \sum_{i=1}^{\ell} b_i A_i) \in \mathbb{Z}_q^{n\times 2m}$.
6. Pick ℓ uniformly random matrices $R_i \in \{-1, 1\}^{m\times m}$, $i \in \{1, \cdots, \ell\}$, and set

$$R = \sum_{i=1}^{\ell} b_i R_i \in \{-\ell, \cdots, \ell\}^{m\times m}.$$

7. Choose $\mathbf{y}_1, \mathbf{y}_2 \in \overline{\Psi}_\alpha^m$, and compute
$\mathbf{z}_1 = R^T\mathbf{y}_1$, $\mathbf{z}_2 = R^T\mathbf{y}_2 \in \mathbb{Z}_q^m$.
8. Compute

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

9. Compute $\mathbf{c}_5 = H_2(\mathbf{c}_1||\mathbf{c}_2||\mathbf{c}_3||\mathbf{c}_4||\beta||f_0||f_1||\cdots||f_{\beta-1})$.
10. The ciphertext is $\mathsf{CT} = (\mathbf{b}, \beta, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5) \in \{-1, 1\}^\ell \times \mathbb{Z}_q^{2t+4m} \times \{0, 1\}^{\lambda+\tau}$.

Decrypt$(\mathsf{SK}, \mathsf{CT})$: Taking as input private key $\mathsf{SK} = (T_A, T_{A'})$ and a ciphertext $\mathsf{CT} = (\mathbf{b}, \beta, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5)$, the algorithm does:

1. Sample $\mathbf{e} \in \mathbb{Z}_q^{2m\times t} \leftarrow \mathsf{SampleLeft}(A, B + \sum_{i=1}^{\ell} b_i A_i, T_A, U, \sigma)$. Note that $F_1 \cdot \mathbf{e} = U \in \mathbb{Z}_q^{n\times t}$.
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, 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 $\mathbf{m}$.
4. Sample $\mathbf{e}' \in \mathbb{Z}_q^{2m\times t} \leftarrow \mathsf{SampleLeft}(A', B + \sum_{i=1}^{\ell} b_i A_i, T_{A'}, U, \sigma)$. Note that $F_2 \cdot \mathbf{e}' = U \in \mathbb{Z}_q^{n\times t}$.
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, i.e. $\left|w_i - \lfloor\frac{q}{2}\rfloor\right| < \lfloor\frac{q}{4}\rfloor$, output 1, otherwise 0. We then obtain the string $(\mathsf{bin}(\delta)||\mathsf{bin}(f(\delta)))$. Then we retrieve the numbers $\delta \in \mathbb{Z}_q$ and $f(\delta) \in \mathbb{Z}_q$.
7. Compute the following values in order $g_0 = H_1(\mathbf{m}||\mathsf{bin}(\beta))$, $g_1 = H_1(\mathbf{m}||\mathsf{bin}(\beta)||g_0)$, $\cdots$, $g_{\beta-1} = H_1(\mathbf{m}||\mathsf{bin}(\beta)||g_0||\cdots||g_{\beta-2})$.
8. For all $x \in \mathbb{Z}_q$, let

$$g(x) = g_0 + g_1 x + \cdots + g_{\beta-1}x^{\beta-1} \in \mathbb{Z}_q.$$

9. Finally, check if $f(\delta) = g(\delta)$ and $\mathbf{c}_5 = H_2(\mathbf{c}_1||\mathbf{c}_2||\mathbf{c}_3||\mathbf{c}_4||\beta||g_0||g_1||\cdots||g_{\beta-1})$ then output $\mathbf{m}$, otherwise output $\perp$.

Aut(SK): On input a private key $\mathsf{SK} = (T_A, T_{A'})$, the authorization algorithm returns the token $\mathsf{TK} = T_{A'}$.

Test($\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma$): Taking as input γ ciphertexts $\mathsf{CT}_i = (\mathbf{b}_i, \beta_i, \mathbf{c}_{i1}, \mathbf{c}_{i2}, \mathbf{c}_{i3}, \mathbf{c}_{i4}, \mathbf{c}_{i5})$ and γ corresponding tokens TK_i. For $i \in \{1, \cdots, \gamma\}$, the test algorithm checks whether the equation $\beta_1 = \cdots = \beta_\gamma = \gamma$ hold or not. If not, it returns $\perp$, otherwise it performs the following:

1. For each $i \in \{1, \cdots, \gamma\}$, do:
   - Note that $\mathbf{b}_i = (b_{i1}, \cdots, b_{i\ell}) \in \{-1, 1\}^\ell$.
   - Sample $\mathbf{e}'_i \in \mathbb{Z}_q^{2m \times t} \leftarrow \mathsf{SampleLeft}(A'_i, B_i + \sum_{k=1}^{\ell} b_{ik} A_{ik}, T_{A'_i}, U_i, \sigma)$. 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 1 if they are close, and 0 otherwise. We obtain the string $(\mathsf{bin}(\delta_i) || \mathsf{bin}(f_i(\delta_i)))$ then we retrieve number δ_i and $f_i(\delta_i) \in \mathbb{Z}_q$.
2. Recall that for all $i \in \{1, \cdots, \gamma\}$,
$$f_i(\delta_i) = f_{i,0} + f_{i,1}\delta_i + f_{i,2}\delta_i^2 + \cdots + f_{i,\gamma-1}\delta_i^{\gamma-1} \in \mathbb{Z}_q.$$
3. With γ pairs of $(\delta_i, f_i(\delta_i))$, we have the following equation system:
$$\begin{cases} f_1(\delta_1) = f_{1,0} + f_{1,1}\delta_1 + \cdots + f_{1,\gamma-1}\delta_1^{\gamma-1} \in \mathbb{Z}_q \\ \qquad \vdots \\ f_\gamma(\delta_\gamma) = f_{\gamma,0} + f_{\gamma,1}\delta_\gamma + \cdots + f_{\gamma,\gamma-1}\delta_\gamma^{\gamma-1} \in \mathbb{Z}_q. \end{cases}$$
   Assume that $f_{i,k} = f_{j,k} = \hat{f}_k$ for $i, j \in \{1, \cdots, \gamma\}$ and $k \in \{0, \cdots, \gamma - 1\}$, we can solve the above equation system and obtain the unique solution $\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_{\gamma-1}$.
4. Then for each $i \in \{1, \cdots, \gamma\}$, check if the following equations hold:
$$\mathbf{c}_{i5} = H_2(\mathbf{c}_{i1} || \mathbf{c}_{i2} || \mathbf{c}_{i3} || \mathbf{c}_{i4} || \gamma || \hat{f}_0 || \hat{f}_1 || \cdots || \hat{f}_{\gamma-1}).$$
   If all the equations hold, return 1, otherwise return 0.

3.2 Correctness of PKE-MET

Theorem 4. *The proposed PKE-MET construction above is correct if H_1 is a one-way hash function and H_2 is a collision-resistant hash function.*

Proof. We go through the following steps for analyzing the proposed PKE-MET construction:

1. In the **Decrypt** algorithm, we have that if CT is a valid ciphertext of $\mathbf{m}$ then by computing $\mathbf{w} \leftarrow \mathbf{c}_1 - \mathbf{e}^T \mathbf{c}_3 \in \mathbb{Z}_q^t$, we get back the message $\mathbf{m}$. Furthermore, it is obvious that the message $\mathbf{m}$ and designated number β satisfy the equalities $f(\delta) = g(\delta)$ and $\mathbf{c}_5 = H_2(\mathbf{c}_1 || \mathbf{c}_2 || \mathbf{c}_3 || \mathbf{c}_4 || \beta || g_0 || g_1 || \cdots || g_{\beta-1})$. The decryption then always returns the message $\mathbf{m}$, that is,
$$\Pr[\mathsf{Dec}(\mathsf{SK}, \mathsf{Enc}(\mathsf{PK}, \mathbf{m}, \beta)) = \mathbf{m}] = 1.$$

2) In the **Test** algorithm, consider the case $\mathbf{m}_1 = \mathbf{m}_2 = \cdots = \mathbf{m}_\gamma$. It is correct to assume that $f_{i,k} = f_{j,k} = \hat{f}_k$ for $i, j \in \{1, \cdots, \gamma\}$ and $k \in \{0, \cdots, \gamma - 1\}$. The equation set then contains γ equations and γ variables. The coefficients forms a Vandermonde matrix as

$$V = \begin{bmatrix} 1 & \delta_1 & \delta_1^2 & \cdots & \delta_1^{\gamma-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \delta_\gamma & \delta_\gamma^2 & \cdots & \delta_\gamma^{\gamma-1} \end{bmatrix} \in \mathbb{Z}_q^{\gamma \times \gamma}.$$

In the Vandermonde matrix, the equation system has a unique solution if $\det(V) = \prod_{1 \le i < j \le \gamma}(\delta_i - \delta_j) \neq 0$. Recall that we randomly choose $\delta_i \in \mathbb{Z}_q$ for $i \in \{1, \cdots, \gamma\}$, thus $\det(V) \neq 0$ with overwhelming probability $\frac{1}{q(q-1)\cdots(q-\gamma+1)}$.
We then solve the matrix and obtain a unique solution $\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_{\gamma-1}$. Since $\mathbf{m}_1 = \mathbf{m}_2 = \cdots = \mathbf{m}_\gamma$, we have that our assumption is correct. Therefore, the unique solution satisfies the following equalities for $i \in \{1, \cdots, \gamma\}$:

$$\mathbf{c}_{i5} = H_2(\mathbf{c}_{i1}||\mathbf{c}_{i2}||\mathbf{c}_{i3}||\mathbf{c}_{i4}||\gamma||\hat{f}_0||\hat{f}_1||\cdots||\hat{f}_{\gamma-1}).$$

Therefore, the **Test** algorithm outputs

$$\mathsf{Test}(\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma) = 1$$

with overwhelming probability.

3) In the **Test** algorithm, assume that $\mathbf{m}_1 = \mathbf{m}_2 = \cdots = \mathbf{m}_\gamma$ does not hold. Then the same argument as 2) can be applied. Without the loss of generality, we can assume that $f_{1,k} \neq f_{i,k} = f_{j,k} = \hat{f}_k$ for $i, j \in \{2, \cdots, \gamma\}$ and $k \in \{0, 1, \cdots, \gamma - 1\}$. The Vandermonde matrix has $\det(V) \neq 0$ with overwhelming probability. However, the Vandermonde matrix's unique solution $\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_{\gamma-1}$ does not satisfy both equations below.

$$\mathbf{c}_{15} = H_2(\mathbf{c}_{11}||\mathbf{c}_{12}||\mathbf{c}_{13}||\mathbf{c}_{14}||\gamma||\hat{f}_0||\hat{f}_1||\cdots||\hat{f}_{\gamma-1}).$$

$$\mathbf{c}_{25} = H_2(\mathbf{c}_{21}||\mathbf{c}_{22}||\mathbf{c}_{23}||\mathbf{c}_{24}||\gamma||\hat{f}_0||\hat{f}_1||\cdots||\hat{f}_{\gamma-1}).$$

Therefore, the **Test** algorithm outputs

$$\mathsf{Test}(\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma) = 0$$

with overwhelming probability.

□

3.3 Security analysis

We show that our proposed PKE-MET construction is OW-CPA secure against Type-I adversaries (cf. Theorem 5), and IND-CPA secure against Type-II adversaries (cf. Theorem 6). Furthermore, we are using the Vandermonde matrix in

the equality test algorithm to secure the information of ciphertexts such that with designated number β, without a sufficient number of CT_γ ciphertexts (e.g., $\gamma \geq \beta$), the equality test cannot be performed. This information-theoretical secure property of our scheme is discussed in Theorem 7.

Theorem 5. *Provided that H_1 is a one-way hash function and H_2 is a collision-resistant hash function. Suppose there exists a probabilistic algorithm $\mathcal{A}$ that wins the OW-CPA game with probability ϵ. Then there is a probabilistic algorithm $\mathcal{B}$ that solves the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE problem with probability*

$$\epsilon_{LWE} \geq \epsilon.$$

Proof. The proof proceeds in a sequence of games where the first game is the original OW-CPA one. In the last game, the challenge ciphertext is chosen randomly. Hence, the advantage of the adversary $\mathcal{A}$ in the last game is zero. Finally, we give an LWE reduction between the last two games.

Let $\mathcal{W}_i$ denotes the event that $\mathcal{A}$ wins Game i. Our goal is to prove that $\Pr[\mathcal{W}_0]$ is negligible. To achieve that, we will show $|\Pr[\mathcal{W}_{i+i}] - \Pr[\mathcal{W}_i]|$ is negligible. We denote the adversary $\mathcal{A}$'s target user is $\mathcal{U}_\theta$ and the challenge ciphertext is $\mathsf{CT}^*_\theta = (\mathbf{b}^*, \beta^*, \mathbf{c}^*_1, \mathbf{c}^*_2, \mathbf{c}^*_3, \mathbf{c}^*_4, \mathbf{c}^*_5)$.

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

Game 1. This is similar to Game 0 except that $\mathcal{C}$ guesses a specific target vector $\mathbf{b}^*$ corresponding to the target CT^*_θ and ℓ random scalars for abort check. At the setup phase, the challenger $\mathcal{C}$ guesses a target vector $\mathbf{b}^* = (b_1, \cdots, b_\ell) \in \mathbb{Z}_q$ and ℓ random scalars $h_i \in \mathbb{Z}_q$ for $i = 1, \cdots, \ell$ such that $1 + \sum_{i=1}^{\ell} b^*_i h_i = 0$. The rest of Game 1 is unchanged with $\mathbf{b}^*$ being used to generate the challenge ciphertext CT^*_θ. In the adversary's view, Game 1 is the same as Game 0. Therefore, we have $\Pr[\mathcal{W}_1] = \Pr[\mathcal{W}_0]$.

Game 2. This is identical to Game 1 except when $\mathcal{C}$ generate the public key for the target user $\mathcal{U}_\theta$. Recall that for each user, the challenger generates public key PK 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 $i \in \{1, \cdots, \ell\}$ be the ephemeral random matrices generated when creating the challenge ciphertext CT^*_θ. At the setup phase, the challenger $\mathcal{C}$ chooses ℓ uniform random matrices R^*_i. It then generates A, A', B as in Game 1 and constructs the matrices A_i, $i \in \{1, \cdots, \ell\}$ as

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

Using [1, Lemma 13], we prove that the matrices A_i are statistically close to uniform and mutually independent. Then we have $\Pr[\mathcal{W}_2] = \Pr[\mathcal{W}_1]$.

Game 3. Game 3 is the same as Game 2, except that we add an abort that is independent of adversary's view as follow:

- (**Abort check**) Whenever adversary $\mathcal{A}$ makes a query to $\mathcal{O}^{\mathsf{Enc}}$ which generates identity vector $\mathbf{b}$, the challenger $\mathcal{C}$ checks if $1 + \sum_{i=1}^{\ell} b_i h_i \neq 0$ with $i \in \{1, \cdots, \ell\}$ where h_i are selected as in Game 1 and kept private by the

challenger $\mathcal{C}$. If not then $\mathcal{C}$ aborts the game. Note that this is unnoticed from the adversary's view and $\mathcal{C}$ can even abort the game as soon as the condition is true.
– (**Artificial abort**) In the final guess phase, the adversary outputs a guess $\mathbf{m}'$ for $\mathbf{m}$. $\mathcal{C}$ samples a random bit Γ such that $\Pr[\Gamma = 1] = \mathcal{G}(\text{all } \mathcal{A}\text{'s queries})$ where $\mathcal{G}(\cdot)$ is a function defined as in [1, Lemma 28]. If $\Gamma = 1$, $\mathcal{C}$ overwrites $\mathbf{m}'$ with a fresh random message and makes a artificial abort.

We can see that in this game, the adversary's view is unchanged. Therefore, we have $\Pr[\mathcal{W}_3] = \Pr[\mathcal{W}_2]$.

Game 4. In this game, we choose A is a uniform random matrix in $\mathbb{Z}_q^{n\times m}$. However B and T_B are generated through $\mathsf{TrapGen}(q, n)$, where T_B is a basis of $\Lambda_q^{\perp}(B)$. The construction of A_i for $i = 1, \cdots, \ell$ remains the same, i.e., $A_i = AR_i^* - h_iB$. The challenger $\mathcal{C}$ then answers the $\mathcal{A}$'s queries via the oracle as follows:

– When $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{SK}}(i \neq \theta)$, the challenger $\mathcal{C}$ returns the generated secret keys SK_i.
– When $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{Aut}}(i)$, the challenger $\mathcal{C}$ returns the corresponding generated trapdoor $T_{A_i'}$ (even in the case $i = \theta$).

The rest of Game 4 is similar to Game 3. In particular, $\mathcal{C}$ uses the abort check in challenge phase and the artificial abort in guess phase. Then Game 4 and Game 3 are identical in the adversary's view, which means $\Pr[\mathcal{W}_4] = \Pr[\mathcal{W}_3]$.

Game 5. Game 5 is identical to Game 4, except that the challenge ciphertext CT_θ is always chosen uniformly at random. And thus $\mathcal{A}$'s advantage is always zero, i.e. $\Pr[\mathcal{W}_5] = 0$.

The remaining part is to show that $|\Pr[\mathcal{W}_5] - |\Pr[\mathcal{W}_4]| \leq \epsilon_{LWE}$ which is negligible proven by the reduction from LWE .

Reduction from the decisional LWE. Recall that a decisional 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 4 and Game 5, 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 t fresh pairs $(\mathbf{u}_i, d_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ with $j \in \{1, \cdots, t\}$ and m fresh pairs $(\mathbf{a}_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ with $i \in \{1, \cdots, m\}$.

Setup. $\mathcal{B}$ executes $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{Setup}(\lambda)$ for $i \in \{1, \cdots, N\}$. Then $\mathcal{B}$ picks randomly target user $\mathcal{U}_\theta$ and constructs the public key PK_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 $\mathbf{a}_i$ 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. As in Game 1, choose target vector $\mathbf{b}^*$ and ℓ random scalars h_i for $i \in \{1, \cdots, \ell\}$ so that $1 + \sum_{i=1}^{\ell} b_i^* h_i = 0$.

4. As in Game 2, the matrices A_i for $i = 1, \cdots, \ell$ are constructed as $A_i \leftarrow A \cdot R_i^* - h_i \cdot B \in \mathbb{Z}_q^{n \times m}$.
5. As in Game 4, run $\mathsf{TrapGen}(q, \sigma)$ to generate matrices $A', B \in \mathbb{Z}_q^{n \times m}$ and their trapdoor $T_{A'}, T_B$ respectively.
6. Set $\mathsf{PK}_\theta := (A, A', A_1, \cdots, A_\ell, B, U)$.

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

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

Challenge. Now when $\mathcal{A}$ sends $\mathcal{B}$ a target user index θ'. If $\theta' \neq \theta$, then $\mathcal{B}$ aborts the game. Otherwise, $\mathcal{B}$ chooses a random message $\mathbf{m} \in \mathcal{M}$ and a designated number β^* and computes the challenge ciphertext $\mathsf{CT}_\theta^* = (\mathbf{b}^*, \beta^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*, \mathbf{c}_4^*, \mathbf{c}_5^*)$ for message $\mathbf{m}$ as follows:

1. Compute the following values in order
$f_0 = H_1(\mathbf{m}||\mathsf{bin}(\beta^*)),\ f_1 = H_1(\mathbf{m}||\mathsf{bin}(\beta^*)||f_0), \cdots, f_{\beta-1} = H_1(\mathbf{m}||\mathsf{bin}(\beta^*)||f_0|| \cdots ||f_{\beta-2})$.
2. For all $x \in \mathbb{Z}_q$, let $f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{\beta-1} x^{\beta-1} \in \mathbb{Z}_q$.
3. Pick randomly number $\delta \in \mathbb{Z}_q$ and compute $f(\delta) \in \mathbb{Z}_q$.
4. Assemble $\mathbf{d}^* = [d_1, \cdots, d_t]^T \in \mathbb{Z}_q^t$, set

$$\mathbf{c}_1^* = \mathbf{d}^* + \mathbf{m}\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t.$$

5. Pick randomly $\mathbf{s}_2 \in \mathbb{Z}_q^n$ and $\mathbf{x}_2 \leftarrow \overline{\Psi}_\alpha^t$, compute

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

6. Compute $R^* = \sum_{i=1}^{\ell} b_i^* R_i^* \in \{-\ell, \cdots, \ell\}^{m \times m}$ with $\mathbf{b}^*$, h_i, and R_i^* for $i \in \{1, \cdots, \ell\}$ are in setup phase.
7. Assemble $\mathbf{v}^* = [v_1, \cdots, v_m]^T \in \mathbb{Z}_q^m$, set

$$\mathbf{c}_3^* = \begin{pmatrix} \mathbf{v}^* \\ (R^*)^T \mathbf{v}^* \end{pmatrix} \in \mathbb{Z}_q^{2m}.$$

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

$$\mathbf{c}_4^* = \begin{pmatrix} A'^T \mathbf{s}_2 + \mathbf{y}_2 \\ (AR^*)^T \mathbf{s}_2 + (R^*)^T \mathbf{y}_2 \end{pmatrix} \in \mathbb{Z}_q^{2m}.$$

9. Compute $\mathbf{c}_5 = H_2(\mathbf{c}_1^*||\mathbf{c}_2^*||\mathbf{c}_3^*||\mathbf{c}_4^*||\beta^*||f_0||f_1|| \cdots ||f_{\beta-1})$.

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

We argue that when the LWE oracle is pseudorandom (i.e. $\mathcal{O} = \mathcal{O}_s$) then CT_θ^* is distributed exactly as in Game 4. It suffices to argue only in case of no abort. We have that
$F_1 := (A|B + \sum_{i=1}^{\ell} b_i^* A_i) = (A|AR^* + h^* B)$
where $R^* \leftarrow \sum_{i=1}^{\ell} b_i R_i^* \in \mathbb{Z}_q^{m \times m}, h^* \leftarrow 1 + \sum_{i=1}^{\ell} b_i^* h_i \in \mathbb{Z}_q$. Since in case of no abort, we have $h^* = 0$ and so $F_1 = (A|AR^*)$. Because the sampling oracle

is pseudorandom, then we have $\mathbf{v}^* = A^T\mathbf{s}_1 + \mathbf{y}_1$ for some random noise vector $\mathbf{y}_1 \leftarrow \overline{\Psi}_\alpha^m$. Therefore, $\mathbf{c}_3^*$ in Step 7 satisfies:

$$\mathbf{c}_3^* := \begin{pmatrix} A^T\mathbf{s}_1 + \mathbf{y}_1 \\ (R^*)^T(A^T\mathbf{s}_1 + \mathbf{y}_1) \end{pmatrix} = (F_1)^T\mathbf{s}_1 + \begin{pmatrix} \mathbf{y}_1 \\ (R^*)^T\mathbf{y}_1 \end{pmatrix}.$$

Moreover, we have $\mathbf{d}^* = U^T\mathbf{s}_1 + \mathbf{x}_1$ for $\mathbf{x}_1 \leftarrow \overline{\Psi}_\alpha^t$ and random $\mathbf{s}_1 \in \mathbb{Z}_q^n$, hence $\mathbf{c}_1^* = U^T\mathbf{s}_1 + \mathbf{x}_1 + \mathbf{m}\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q^t$. 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 $\mathbf{c}_1^*$ is uniform. It follows also from the leftover hash lemma (cf. [22, Theorem 8.38]) that $\mathbf{c}_3^*$ is also uniform in $\mathbb{Z}_q^{2m}$. Consequently, the challenge ciphertext CT_θ^* is always uniform in $\{-1,1\}^\ell \times \mathbb{Z}_q^{2t+4m} \times \{0,1\}^{\lambda+\tau}$.

Guess. After making additional queries in Phase 2, $\mathcal{A}$ guesses if it is interacting with Game 4 or Game 5. The simulator also implements the artificial abort for these Games and uses the final guess to answer the decisional LWE problem.

We claim 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 same as in Game 5. Hence $\mathcal{B}$'s advantage in solving the LWE problem is the same as the advantage of $\mathcal{A}$ in distinguishing Game 4 and Game 5. Accumulate all the probability equations above, and we have that $\Pr[\mathcal{W}_0] = \epsilon \leq \epsilon_{LWE}$ as required.

□

Theorem 6. *Provided that H_1 is a one-way hash function and H_2 is a collision-resistant hash function, and the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumption holds. Suppose there exists a probabilistic algorithm $\mathcal{A}$ that wins the IND-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 full proof is similar to that of Theorem 5. The proof proceeds in a sequence of games where the first game is the original IND-CPA one. In the last game, the challenge ciphertext is chosen randomly. Hence, the success probability of the adversary $\mathcal{A}$ in the last game is zero. Finally, we give an LWE reduction between the last two games.

The goal is to show our construction is indistinguishable from random, meaning the challenge ciphertext is indistinguishable from a random element in the ciphertext space. We denote the adversary $\mathcal{A}$'s target user is $\mathcal{U}_\theta$ and the challenge ciphertext is $\mathsf{CT}_\theta^* = (\mathbf{b}^*, \beta^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*, \mathbf{c}_4^*, \mathbf{c}_5^*)$.

The serial IND-CPA games are similar to OW-CPA games. Except that i) In **Game 3**, when abort happens, $\mathcal{C}$ overwrites b' with a fresh random bit in $\{0,1\}$ and aborts the game. ii) In **Game 4**, when abort happens, $\mathcal{C}$ does not need to answer the query $\mathcal{O}^{\mathsf{Aut}}$ for target user $\mathcal{U}_\theta$. iii) In **Challenge** phase, $\mathcal{A}$ sends $\mathcal{B}$ two messages $\mathbf{m}_0$ and $\mathbf{m}_1$ of the same length and a target user index θ'. Challenger $\mathcal{C}$ chooses a random bit $b \in \{0,1\}$, a designated number β^*, and computes the challenge ciphertext $\mathsf{CT}_\theta^* = (\mathbf{b}^*, \beta^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*, \mathbf{c}_4^*, \mathbf{c}_5^*)$ for $\mathbf{m}_b$ then send CT_θ^* to $\mathcal{A}$.

□

3.4 Number Security of PKE-MET

Recall that the citizen gives their consent for the Health Officer in our cloud model by sending their tokens. The trusted Health Officer could only compute the statistical data against ciphertexts that have the same designated number β. Furthermore, the number of ciphertexts that will be computed must be equal to or larger than the designated number. Suppose that an attacker has the token of γ users whose ciphertexts are associated with the designated number β where $\gamma < \beta$. Now the attacker attempts to find out whether these ciphertexts have the same underlying message or not. In the following theorem, we will prove that it is impossible for the attacker to perform an equality test in this case.

Theorem 7. *In the information-theoretical sense, the proposed* PKE-MET *scheme is secure against the number game which is defined in Section 2.*

Proof. Recall that an attacker $\mathcal{A}$ has the token of γ users whose ciphertexts are CT_i for $i \in [\gamma]$ with the designated number β where $\gamma < \beta$. Hence, there are two cases that the attacker $\mathcal{A}$ can find out if the ciphertexts CT_i have the same underlying message or not:

1. Case 1: $\mathcal{A}$ tries to extract the corresponding underlying message from each given ciphertext CT_i with $i \in \{1, \cdots, \gamma\}$.
2. Case 2: $\mathcal{A}$ tries to extract the coefficient values $f_{i,j}$ from the Vandermonde matrix with the given ciphertexts CT_i with $i \in \{1, \cdots, \gamma\}$.

For Case 1. According to the OW-CPA security proof in Theorem 5, the attacker $\mathcal{A}$ has no better chance than guessing when trying to obtain the underlying messages from given ciphertexts.

For Case 2. To extract the coefficient values $f_{i,j}$ from given ciphertexts $\mathsf{CT}_i = (\mathbf{b}_i, \beta_i, \mathbf{c}_{i1}, \mathbf{c}_{i2}, \mathbf{c}_{i3}, \mathbf{c}_{i4}, \mathbf{c}_{i5})$, $\mathcal{A}$ already has the tokens TK_i on target users $\mathcal{U}_i$ for $i \in [\gamma]$, then $\mathcal{A}$ can use the following procedure (exactly the same as the **Test** algorithm on γ ciphertexts where $\gamma < \beta$) to re-construct the Vandermonde matrix:

1. For each $i \in \{1, \cdots, \gamma\}$, do:
   - From ciphertext, take $\mathbf{b}_i = (b_{i1}, \cdots, b_{i\ell})$.
   - Sample $\mathbf{e}'_i \in \mathbb{Z}_q^{2m \times t}$ from $\mathbf{e}'_i \leftarrow \mathsf{SampleLeft}(A'_i, B_i + \sum_{k=1}^{\ell} b_{ik} A_{ik}, T_{A'_i}, U_i, \sigma)$.
   - 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 1 if they are close, and 0 otherwise. $\mathcal{A}$ obtains the vectors $(\mathsf{bin}(\delta_i)||\mathsf{bin}(f_i(\delta_i)))$.
2. Recall that for all $i \in \{1, \cdots, \gamma\}$, $f_i(\delta_i) = f_{i,0} + f_{i,1}\delta_i + f_{i,2}\delta_i^2 + \cdots + f_{i,\gamma-1}\delta_i^{\gamma-1} \in \mathbb{Z}_q$ where $f_{i,k} = H_1(\mathbf{m}||\mathsf{bin}(\beta)||f_{0,k}|| \cdots ||f_{i-1,k})$ for $i \in \{1, \cdots, \gamma\}$ and $k \in \{0, \cdots, \beta - 1\}$.
3. With γ pair δ_i and $f_i(\delta_i)$ values, $\mathcal{A}$ has the following equation set:

$$\begin{cases} f_1(\delta_1) = f_{1,0} + f_{1,1}\delta_1 + \cdots + f_{1,\beta-1}\delta_1^{\beta-1} \in \mathbb{Z}_q \\ \qquad \vdots \\ f_\gamma(\delta_\gamma) = f_{\gamma,0} + f_{\gamma,1}\delta_\gamma + \cdots + f_{\gamma,\beta-1}\delta_\gamma^{\beta-1} \in \mathbb{Z}_q \end{cases}$$

where δ_i for $i \in \{1, \cdots, \gamma\}$ are chosen randomly.

$\mathcal{A}$ then can assume that $f_{i,k} = f_{j,k} = \hat{f}_k$ for $i, j \in \{1, 2, \cdots, \gamma\}$ and $k \in \{0, 1, \cdots, \beta - 1\}$. This is a set of γ equations with β variables where $\beta > \gamma$. According to the Vandermonde matrix properties, these γ equations are non-linearly correlated with each other by overwhelming probability. Therefore, there are an infinite number of coefficients $\{f_{i,j}\}_{1 \le i \le \gamma, 0 \le j \le \beta - 1}$.

Furthermore, based on the proof of Shamir secret sharing scheme [6], $\mathcal{A}$ has no additional information to determine coefficients $f_{i,j}$ and the equality of underlying messages. Therefore, the proposed PKE-MET scheme is number secure. □

3.5 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 $\|\widetilde{T_A}\|, \|\widetilde{T_B}\| \le O(\sqrt{n \log q})$.
- 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 user 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 Lattice-based PKE-FMET Construction

In previous construction, we see that PKE-MET supports equality test of multiple ciphertexts at once without leaking information during the Test algorithm execution. However, it requires the number of ciphertexts to match the designated number β. This condition seems rather strict and unpractical. To remove that drawback, a new notion of public-key encryption with flexible multi-ciphertext equality test (PKE-FMET) was introduced. The scheme can perform equality tests on ciphertexts which designate different numbers as long as the maximal number is less than or equal to the number of ciphertexts. For instance, given γ ciphertexts $\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma$ in which CT_i designates number β_i. The equality test can be performed among these ciphertexts if $\beta \le \gamma$ for $\beta = \mathsf{max}\{\beta_1, \cdots, \beta_\gamma\}$. A threshold ω is introduced such that an equality test can be performed on maximal ω ciphertexts. Based on the above PKE-MET scheme, one can extend it to several PKE-FMET constructions as follows:

4.1 Trivial PKE-FMET construction

The **Setup**, **KeyGen**, and **Aut** algorithms are the same as PKE-MET construction.

For **Encrypt**$(\mathsf{PK}, \mathbf{m}, \beta)$ algorithm: Since $(\mathbf{c}_2, \mathbf{c}_4, \mathbf{c}_5)$ in the PKE-MET scheme is computed with the designated number β, we add $\omega - \beta$ pairs of $(\mathbf{c}_{2i}, \mathbf{c}_{4i}, \mathbf{c}_{5i})$ for $i \in \{\beta+1, \cdots, \omega\}$ with the same computation as the original $(\mathbf{c}_2, \mathbf{c}_4, \mathbf{c}_5)$. The $(\mathbf{c}_{2i}, \mathbf{c}_{4i})$ are calculated with the same matrix U and different noise $\mathbf{s}_2, \mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2$.

1. For maximal number ω, compute
   $f_0^\omega = H_1(\mathbf{m}||\mathsf{bin}(\omega))$,
   $f_1^\omega = H_1(\mathbf{m}||\mathsf{bin}(\omega)||f_0^\omega)$, $\cdots$,
   $f_{\omega-1}^\omega = H_1(\mathbf{m}||\mathsf{bin}(\omega)||f_0^\omega||\cdots||f_{\omega-2}^\omega)$.
2. Then for $i \in \{\beta, \cdots, \omega-1\}$, compute $f_0^i, f_1^i, \cdots, f_{i-1}^i$ values in order
   $f_0^{\omega-1} = H_1(f_0^\omega)$, $\cdots$, $f_{\omega-2}^{\omega-1} = H_1(f_{\omega-2}^\omega)$,
   $\vdots$
   $f_0^\beta = H_1(f_0^{\beta+1})$, $\cdots$, $f_{\beta-1}^\beta = H_1(f_{\beta-1}^{\beta+1})$.
3. For $i \in \{\beta, \cdots, \omega\}$, the pair values $(\delta, f^i(\delta))$ can be used to compute $(\mathbf{c}_2, \mathbf{c}_4, \mathbf{c}_5)$ as:

   $\mathbf{c}_2^i = U^T\mathbf{s}_2^i + \mathbf{x}_2^i + (\mathsf{bin}(\delta)||\mathsf{bin}(f^i(\delta)))\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$.
   Set $\mathbf{c}_2 = \mathbf{c}_2^\beta||\cdots||\mathbf{c}_2^\omega \in \mathbb{Z}_q^{(\omega-\beta+1)t}$.

   $\mathbf{c}_4^i = F_2^T\mathbf{s}_2^i + \begin{pmatrix}\mathbf{y}_2^i \\ \mathbf{z}_2^i\end{pmatrix} \in \mathbb{Z}_q^{2m}$.
   Set $\mathbf{c}_4 = \mathbf{c}_4^\beta||\cdots||\mathbf{c}_4^\omega \in \mathbb{Z}_q^{2(\omega-\beta+1)m}$.
   $\mathbf{c}_5 = H_2(\mathbf{c}_1||\mathbf{c}_2||\mathbf{c}_3||\mathbf{c}_4||\beta||f_0^\beta||f_1^\beta||\cdots||f_{\beta-1}^\beta)$.

4. The ciphertext is $\mathsf{CT} = (\mathbf{b}, \beta, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5) \in \{-1, 1\}^\ell \times \mathbb{Z}_q^{(\omega-\beta+2)(t+2m)} \times \{0,1\}^{\lambda+\tau}$.

For **Decrypt**$(\mathsf{SK}, \mathsf{CT})$ algorithm: each $\mathbf{c}_2^i$ and $\mathbf{c}_4^i$ have fixed size. We can recover a number $\delta \in \mathbb{Z}_q$ and $f^i(\delta) \in \mathbb{Z}_q$, for $i \in \{\beta, \cdots, \omega\}$.

For **Test**$(\mathsf{CT}_1, \cdots, \mathsf{CT}_\gamma, \mathsf{TK}_1, \cdots, \mathsf{TK}_\gamma)$ algorithm: On input γ ciphertexts $\mathsf{CT}_i = (\mathbf{b}_i, \beta_i, \mathbf{c}_{i1}, \mathbf{c}_{i2}, \mathbf{c}_{i3}, \mathbf{c}_{i4}, \mathbf{c}_{i5})$ and γ corresponding token TK_i where $i \in \{1, \cdots, \gamma\}$, the test algorithm checks if $\beta \le \gamma \le \omega$ for $\beta = \mathsf{max}\{\beta_1, \cdots, \beta_\gamma\}$. If it does not hold, return $\perp$, otherwise it re-construct the Vandermonde matrix, solve it and verifies the correctness of all equations $\mathbf{c}_{i5} = H_2(\mathbf{c}_{i1}||\mathbf{c}_{i2}||\mathbf{c}_{i3}||\mathbf{c}_{i4}||\beta_i||f_{i,0}^{\beta_i}||f_{i,1}^{\beta_i}||\cdots||f_{i,\beta_i-1}^{\beta_i})$.

Discussion This construction keeps the public key size the same as before but sacrifices the ciphertext size and computation efficiency to realize the flexible multi-ciphertext equality test trivially. For instance, the ciphertext size linearly increases when the designated number β is small. It leads to the inapplicability of this construction in practical applications.

4.2 Naive PKE-FMET construction

An improved idea is that we can generate additional ω uniformly random matrices $\{U_i\}_{1\leq i\leq \omega}$ and modify the PKE-MET construction as follows:

The **Setup** and **Aut** algorithms are the same as 4.1.

In the **KeyGen** algorithms: Select $\omega+1$ uniformly random matrices $\{U_i\}_{0\leq i\leq \omega} \in \mathbb{Z}_q^{n\times t}$.

The **Encrypt** algorithm is similar to Section 4.1, except that we compute $\mathbf{c}_2 = (U\|U_\beta\|\cdots\|U_\omega)^T\mathbf{s}_2+\mathbf{x}_2+(\mathsf{bin}(\delta)||\mathsf{bin}(f^\beta(\delta))\|\cdots\|\mathsf{bin}(f^\omega(\delta)))\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q^{(\omega-\beta+1)t}$ and $\mathbf{c}_4$ is the same as PKE-MET construction.

The **Decrypt** and **Test** algorithms are similar to 4.1, except that we compute $\mathbf{e} \leftarrow \mathsf{SampleLeft}(A, B+\sum_{i=1}^{\ell} b_i A_i, T_A, (U\|U_\beta\|\cdots\|U_\omega, \sigma)$.

Discussion This construction sacrifices the public key size and computation efficiency to realize the flexible multi-ciphertext equality test. However, we achieve an acceptable ciphertext size with an increase of $(\omega-\beta)t$ bits comparing to the PKE-MET scheme.

4.3 A better PKE-FMET construction

We can improve the computation efficiency by calculating $\mathbf{c}_2$ with the same U and $\mathbf{s}_2$ while generating different noise value $\mathbf{x}_2$ each iteration i for $i \in \{\beta, \cdots, \omega\}$ the construction is updated as follow:

The **Setup**, **KeyGen** and **Aut** algorithms are similar to 4.1.

The **Encrypt** algorithm is similar to Section 4.2, except that we compute $\mathbf{c}_2$ as

$\mathbf{c}_2' = U^T\mathbf{s}_2 + \mathbf{x}_2 + (\mathsf{bin}(\delta))\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q^t$

$\mathbf{c}_2^i = U^T\mathbf{s}_2 + \mathbf{x}_2^i + (\mathsf{bin}(f^i(\delta)))\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q^t$

Set $\mathbf{c}_2 = \mathbf{c}_2'||\mathbf{c}_2^\beta||\cdots||\mathbf{c}_2^\omega \in \mathbb{Z}_q^{(\omega-\beta+2)t}$.

The **Decrypt** and **Test** algorithms are similar to 4.1.

Discussion This construction has the best storage capability and scalability, in which the public key size does not increase, while ciphertext size only increases with the minimal value. However, we may face some potential problems by calculating $\mathbf{c}_2$ using the same matrix U and secret $\mathbf{s}_2$ multiple times while the noise $\mathbf{x}_2$ is randomly chosen for each computation. The adversary might or might not learn something from the ciphertext. Thus, this idea requires more mathematical analyses.

4.4 Correctness Analysis and Security analysis

They are similar to Section 3.1 so we can omit them.

5 Conclusion

In this paper, we propose a direct construction of PKE-MET based on the hardness of the Learning With Errors problem. We give its security proofs under the

defined security models. Furthermore, we extend the PKE-MET construction and propose the so-called PKE-FMET to be more practical in cloud computing applications.

We list several options for the PKE-FMET construction. They are interesting directions for further enhancements. We also leave as future work improving our schemes to achieve the CCA2-security.

Acknowledgment

In this paper, we propose a direct construction of PKE-MET based on the hardness of Learning With Errors problem. We give its security proofs under the defined security models. Furthermore, we extend the PKE-MET construction and propose PKE with flexible MET to enable it more practical in application. Our PKE-FMET obtains constants size. These constructions inherited the idea from Susilo [23]. The public key size and private key size remained the same. However, the ciphertext size increase, especially in the case of flexible MET. 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.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 553–572, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
2. Miklos Ajtai. Generating hard instances of the short basis problem. In Jiri Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming*, pages 1–9, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
3. Seth Alornyo, Michael Asante, Xiong Hu, and Kingsford Kissi Mireku. Encrypted traffic analytic using identity based encryption with equality test for cloud computing. pages 1–4, 08 2018.
4. Joel Alwen and Chris Peikert. Generating shorter bases for hard random lattices. Cryptology ePrint Archive, Report 2008/521, 2008. https://eprint.iacr.org/2008/521.
5. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe, and compact garbled circuits. Cryptology ePrint Archive, Report 2014/356, 2014. https://eprint.iacr.org/2014/356.
6. Dan Boneh and Victor Shoup. A Graduate Course in Applied Cryptography New Version. *Crypto.Stanford.Edu*, (1):447–449, 2019.
7. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 207–222, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

8. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 523–552, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
9. Khadijah Chamili, Md Jan Nordin, Waidah Ismail, and Abduljalil Radman. Searchable encryption: A review. *International Journal of Security and Its Applications*, 11:79–88, 12 2017.
10. 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 Julian Jang-Jaccard and Fuchun Guo, editors, *Information Security and Privacy*, pages 138–155, Cham, 2019. Springer International Publishing.
11. Dung Hoang Duong, Huy Quoc Le, Partha Sarathi Roy, and Willy Susilo. Lattice-Based IBE with Equality Test in Standard Model. In Ron Steinfeld and Tsz Hon Yuen, editors, *Provable Security*, pages 19–40, Cham, 2019. Springer International Publishing.
12. Dung Hoang Duong, Willy Susilo, Minh Kim Bui, and Thanh Xuan Khuc. A lattice-based certificateless public key encryption with equality test in standard model. In Zhe Liu and Moti Yung, editors, *Information Security and Cryptology*, pages 50–65, Cham, 2020. Springer International Publishing.
13. Huy Quoc Le, Dung Hoang Duong, Partha Sarathi Roy, Willy Susilo, Kazuhide Fukushima, and Shinsaku Kiyomoto. Lattice-based signcryption with equality test in standard model. *Computer Standards & Interfaces*, 76:103515, 2021.
14. 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. https://eprint.iacr.org/2016/1182.
15. Sha Ma. Identity-based encryption with outsourced equality test in cloud computing. *Information Sciences*, 328:389–402, 01 2016.
16. 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*, 10:458–470, 03 2015.
17. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. volume 37, pages 372– 381, 11 2004.
18. Giang Linh Duc Nguyen, Willy Susilo, Dung Hoang Duong, HuyQuoc Le, and Fuchun Guo. Lattice-Based IBE with Equality Test Supporting Flexible Authorization in the Standard Model. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology – INDOCRYPT 2020*, pages 624–643, Cham, 2020. Springer International Publishing.
19. Mohammed Ramadan, Yongjian Liao, Fagen Li, Shijie Zhou, and Hisham Abdalla. Ibeet-rsa: Identity-based encryption with equality test over rsa for wireless body area networks. *Mobile Networks and Applications*, 04 2019.
20. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. volume 56, pages 84–93, 01 2005.
21. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.
22. Igor Shparlinski. A computational introduction to number theory and algebra. *Math. Comput.*, 76:1697–1698, 09 2007.
23. W. Susilo, F. Guo, Z. ZHAO, and G. Wu. Pke-met: Public-key encryption with multi-ciphertext equality test in cloud computing. *IEEE Transactions on Cloud Computing*, pages 1–1, 2020.

24. Libing Wu, Yubo Zhang, Kim-Kwang Raymond Choo, and Debiao He. Efficient identity-based encryption scheme with equality test for smart city. *IEEE Transactions on Sustainable Computing*, PP:1–1, 07 2017.
25. Guomin Yang, Chik How Tan, Qiong Huang, and Duncan S. Wong. Probabilistic public key encryption with equality test. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010*, pages 119–131, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
26. Kai Zhang, Jie Chen, Hyung Tae Lee, Haifeng Qian, and Huaxiong Wang. Efficient public key encryption with equality test in the standard model. *Theoretical Computer Science*, 755:65 – 80, 2019.