IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 June 2024, accepted 4 August 2024, date of publication 16 August 2024, date of current version 28 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3445294

THEORY

Tightly Secure Public Key Encryption With Equality Test in Setting With Adaptive Corruptions

YUNHAO LING
Software Engineering Institute, East China Normal University, Shanghai 200062, China
e-mail: yhlingyy@163.com

This work was supported in part by the National Natural Science Foundation of China under Grant 62372180, and in part by the Innovation Program of Shanghai Municipal Education Commission under Grant 2021-01-07-00-08-E00101.

ABSTRACT Public Key Encryption with Equality Test (PKEET) is a cryptographic primitive that allows an authorized entity to test whether two given ciphertexts are the encryption of the same message without decrypting them. The security of cryptographic schemes is analyzed using security model, and thus in order to derive reasonable security against the real attackers, the security model should reflect the real attack as closely as possible. However, security model widely used by PKEET fails to capture corruption attack, since it does not cover the real attacker who can adaptively corrupt users. On the other hand, many PKEET schemes suffer from a security loss that is linear in the number of users when using security model with adaptive corruption attack, which causes that the actual security guarantees of the schemes linearly degrade in that. Therefore, the goal of this paper is to resolve these two problems. We present a PKEET scheme in setting with adaptive corruptions in which the security loss is a constant, and in particular, the comparison shows that our scheme is efficient.

INDEX TERMS Multi-user setting, adaptive corruptions, tight reduction, public key encryption, equality test.

I. INTRODUCTION

Public Key Encryption with Equality Test (PKEET) [1] is a cryptographic primitive that allows an authorized entity to test whether two given ciphertexts are the encryption of the same message *without decrypting them*. More specifically, assume that ct_A and ct_B be the encryption of message m_A under Alice's key pk_A and the encryption of the message m_B under Bob's key pk_B, respectively. An entity, with the permission of Alice and Bob, can decide by running the Test algorithm whether ct_A and ct_B are the encryption of the same message, namely, whether or not $\mathsf{m}_A = \mathsf{m}_B$. Such equality test functionality is very useful, and thus PKEET becomes a tool for providing and enhancing privacy in a variety of settings from encrypted database [1] to cloud computing [2], outsourced private set [3], and smart grid [4].

Over the past decade, much progress has been made on the design and analysis of PKKETs, leading to many theoretical achievements, for example, the construction in the standard model [5], [6], [7], the construction against quantum adversaries [8], [9], the construction against inside adversaries [10], the generic construction [11] and strong security [12]. However, there are still fundamental problems needed to resolve.

A. MOTIVATION

Specifically, we focus on the following problems.

1) COVERING REAL CORRUPTION ATTACK

The security of cryptographic schemes is analyzed using security model, and thus in order to derive reasonable security

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

against the real attackers, the security model should reflect the real attack as closely as possible. However, security model widely used by PKEET fails to capture corruption attack, defined as follows.

- *The adversary must declare a target user before seeing any parameter.* (2) The challenger generates μ public/secret keys and sends those public keys to the adversary. (3) The adversary can make a series of queries, including the corrupted key query in which it can get many users' secret key *except the target user's secret key*. (4) The adversary outputs the guess.

Observe that the adversary can make corrupted key query, and indeed, the corruption attack should be taken into account. However, the description of corruption attack from the real attacker is far from sufficient. As to the real attacker, it could reveal secret keys of some users (probably through hack attack or bad key management), and in particular, it can choose the target *at any time*. That means that it is unreasonable to require the adversary to choose the target before the system is built, as the security model above. In fact, *adaptive corruption* in which the adversary can adaptively choose the target users and corrupted secret keys *at any time*, is an important feature to define security model in multi-user setting [13], [14]. Some multi-user cryptosystems, for example, proxy re-encryption [15], [16], [17], has discussed adaptive corruptions and the design of constructions against adaptive corruptions [18], [19]. Henceforth we say that a scheme is *adaptively secure* if the corruption of the adversary is adaptive, and otherwise selectively secure.

2) OBTAINING TIGHT SECURITY REDUCTION

The security of the scheme is proven by designing a reduction algorithm which converts a $(t_{\mathcal{A}}, \epsilon_{\mathcal{A}})$-adversary against the scheme into an efficient $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-algorithm against the computational hard problem, and the scheme is secure if the computational hard problem holds. In general, $t_{\mathcal{A}} \approx t_{\mathcal{B}}$, and we only concern on $\epsilon_{\mathcal{A}}$ and $\epsilon_{\mathcal{B}}$. The security loss is defined by $L = \epsilon_{\mathcal{A}}/\epsilon_{\mathcal{B}}$, and the scale of L reflects the gap between security level of the scheme and hardness of the computational hard problem. A tight security reduction is one where L is a constant. We say a scheme is tightly secure if the security reduction is tight.

The difficult in constructing a PKEET scheme proven in setting with adaptive corruptions is to derive tight security reductions. Actually, a PKEET scheme that is selectively secure is also adaptively secure: the proof of adaptive security can be reduced to selective security by initially guessing the targe user with a successful probability $1/\mu$. However, this reduction always suffers from a security loss of μ, and therefore the loss is linear in the number of users. Consequently, the actual security guarantees of the scheme degrade linearly in that, which can be potential problems:

- If a PKEET system has a huge number of users, the security guarantees could degrade heavily. For instance, there are 2^{30} users, which is possible, imagining billions of users over mobile Internet. When choosing the security parameter providing, for example, 128-bit security level, the *actual* security guarantees is only 98-bit security level, which is insufficient. One may select the security parameter providing 158-bit security level in order for actual 128-bit security level. However, the large security parameters result in the large size of the underlying groups, and accordingly increase the running time of the implementation [20].
- On the other hand, if the number of actual users grows beyond 2^{30}, the security guarantees will be less than 128-bit security level, and we have to reinitialize the system, which is unrealistic, of course.

Therefore, the goal of this paper is to address the two problems above. Concretely, we aim to designs a PKEET scheme in which the security model defines *adaptive* corruption attack and the security reduction should be *tight*.

B. OUR RESULTS

We present a *tightly secure* PKEET scheme in setting with *adaptive corruptions*. We note that in our security model, the adversary can adaptively corrupt users and is not needed to submit any target user before seeing parameters, which models the real-world attacker. Besides, our scheme achieves tighter security reduction, obtaining better security guarantees in practice as well as better results in theory.

Table 1 gives comparison of Efficiency and Feature between our scheme and related PKEET schemes. Here we choose PKEET schemes designed by Tang [21], Ma et al. [22], Zhang et al. [5], Zeng et al. [6], denoted by Tan12, MZH+15, ZCL+19, ZCZ+19, respectively. We need to point out that our scheme is CPA secure, and other schemes are CCA secure. In fact, CPA security is sufficient for actual use. From the table, we can conclude that our scheme is efficient, compared to other schemes, and most importantly, our scheme can achieve tightly adaptive security.

Our Techniques. Firstly, let us explain why many PKEET schemes fail to obtain tight security reductions in proving adaptive security. Generally, in PKEET schemes, each user has a public/secret key pair formed as

$$(\mathsf{pk}, \mathsf{sk}) = ((g^x, g^y), (x, y)),$$

Without loss of generality, we suppose that the parameter of the instance of the hard problem is embedded in g^x. Specifically, given an instance (g^u, g^v, T) of CDH/DDH problem, the simulator does not know which user the adversary will choose as the target, so it randomly chooses an expected user i and generates the public key pk_i as follow.

1) Choose $y^{(i)} \xleftarrow{\$} \mathbb{Z}_q$, and compute $g^{y^{(i)}}$;
2) Set $g^{x^{(i)}} = g^u$.

We note that u is unknown to the simulator, and thus the simulator cannot answer this user's secret key sk_i. The reduction succeeds if the adversary

- never requests the secret key sk_i;
- chooses the expected user i as the target.

TABLE 1. Comparison of Efficiency and Feature between our scheme and related PKEET schemes. Column |pk|, |sk|, |td|, |ct| show the size of public keys, secret keys, trapdoors and ciphertexts, respectively. Column T_{Enc}, T_{Aut} and T_{Test} show encryption cost, trapdoor generation cost and test cost, respectively. $\tilde{E}$ and E refer to exponentiations on group $\mathbb{G}$ with pairings and exponentiations on group G without pairings, respectively. P refers to pairings.

	$\|\mathsf{pk}\|$	$\|\mathsf{sk}\|$	$\|\mathsf{td}\|$	$\|\mathsf{ct}\|$	T_{Enc}	T_{Aut}	T_{Test}	Tightly Adaptive Security
Tan12	$2G$	$2\|\mathbb{Z}_q\|$	$\|\mathbb{Z}_q\|$	$4G+\|\mathbb{Z}_q\|+2\lambda$	$5E$	$2E$	$4E$	No
MZH+15	$3\|\mathbb{G}\|$	$3\|\mathbb{Z}_q\|$	$\|\mathbb{Z}_q\|$	$5\|\mathbb{G}\|+\|\mathbb{Z}_q\|$	$6\tilde{E}$	$5\tilde{E}$	$5\tilde{E}+2P$	No
ZCL+19	$3\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+3\|\mathbb{Z}_q\|$	$\|\mathbb{G}\|$	$2\|\mathbb{G}\|+2\|\mathbb{G}_T\|+\|\mathbb{Z}_q\|$	$6\tilde{E}$	$1\tilde{E}+2P$	$2P$	No
ZCZ+19	$4\|G\|$	$8\|\mathbb{Z}_q\|$	$2\|\mathbb{Z}_q\|$	$5\|G\|$	$4E$	$4E$	$4E$	No
Ours	$2\|G\|$	$4\|\mathbb{Z}_q\|$	$2\|\mathbb{Z}_q\|$	$4\|G\|$	$4E$	$2E$	$4E$	**Yes**

Since the probability of successful reduction is $1/\mu$, the reduction loss is at least μ. Hence this proof strategy inherently suffers from a loss of $\mathcal{O}(\mu)$.

Now we give our solution. The key point is to avoid the guess. Our PKEET scheme is presented as follows.

$$\begin{aligned}\mathsf{pp} &:= (q, G, g, g^a, \mathsf{H}),\\ (\mathsf{pk}, \mathsf{sk}) &:= ((g^{x_1+ax_2}, g^{y_1+ay_2}), (x_1, x_2, y_1, y_2)),\\ \mathsf{ct} &:= (g^r, g^{ar}, g^{(x_1+ax_2)r}\cdot \mathsf{H}(\mathsf{m}), g^{(y_1+ay_2)r}\cdot \mathsf{m}),\\ \mathsf{td} &:= (x_1, x_2).\end{aligned}$$

At a high level, in our proof, the simulator generates and can know any secret key of users, and the reduction is always successful regardless of which user will be the target.

Proving OW-CPA security against Type-I adversary. The goal of this adversary is to recover the message in the ciphertext. In the first step, we convert ct into ct' which has the following form.

$$\mathsf{ct}' = (g^r, \boxed{g^{r'}}, \boxed{g^{x_1r+x_2r'}}\cdot \mathsf{H}(\mathsf{m}), \boxed{g^{y_1r+y_2r'}}\cdot \mathsf{m}),$$

Note that we draw boxes to highlight the difference. Intuitively, this should follow from the DDH assumption, which says that $\{g^a, g^r, g^{ar}\} \approx_c \{g^a, g^r, g^{r'}\}$. Note that given (g^u, g^v, T), we have the following setting.

$$g^a = g^u, \quad g^r = g^v, \quad g^{ar} = T.$$

Apparently, no parameter of (g^u, g^v, T) is embedded in public keys. Thus, the simulator generates and can know all secret key, and is able to return any user's secret key. Furthermore, the setting of T is independent from the target user, in other words, the adversary must answer the hard problem, no matter which user is selected as the target. Thus the reduction is always successful.

In the second step, we use information-theoretic arguments to prove that $g^{y_1r+y_2r'}$ is a perfect one-time pad, so we can replace the message m with a random message m_R, namely, we can convert ct' into ct'' which has the following form.

$$\mathsf{ct}' = (g^r, g^{r'}, g^{x_1r+x_2r'}\cdot \mathsf{H}(\mathsf{m}), g^{y_1r+y_2r'}\cdot \boxed{\mathsf{m}_R}).$$

To see this, given

$$y_1 + ay_2$$

from the public key,

$$y_1r + y_2r'$$

from the ciphertext is uniformly distributed from the adversary's view, since y_1, y_2 are picked at random over $\mathbb{Z}_q^2$, and in addition, the determinant

$$\begin{vmatrix} 1 & a \\ r & r' \end{vmatrix} \neq 0$$

and the solution is unique. Finally, as to $\mathsf{H}(\mathsf{m})$, the adversary cannot recover m from W due to the one-wayness of the hash function.

Note that in this step, we do not employ any computational assumption, and thus the simulator generates and can know all secret key, and is able to return any user's secret key. In addition, for any target user i^* chosen by the adversary, the following two distributions are statistically identical:

$$\left\{y_1^{(i^*)} + ay_2^{(i^*)}, y_1^{(i^*)}r + y_2^{(i^*)}r'\right\} \text{ and } \left\{y_1^{(i^*)} + ay_2^{(i^*)}, z\right\},$$

where $z \stackrel{\$}{\leftarrow} \mathbb{Z}_q$. That means that we can always mask the message regardless of which user will be the target.

Proving IND-CPA security against Type-II adversary. The goal of this adversary is to decide the ciphertext is the encryption of which message. In the first step, we convert ct into ct' which has the following form.

$$\mathsf{ct}' = (g^r, g^{r'}, g^{x_1r+x_2r'}\cdot \mathsf{H}(\mathsf{m}_\beta), g^{y_1r+y_2r'}\cdot \mathsf{m}_\beta),$$

This step is completely analogue to the above.

In the second step, we use information-theoretic arguments to prove that $g^{y_1r+y_2r'}$ is a perfect one-time pad, and $g^{x_1r+x_2r'}$ is also a perfect one-time pad. These are analogue to the above. Thus, β is independent from the adversary's view.

C. RELATED WORK

The concept of PKEET was proposed by Yang et al. [1].

Later, Tang et al. [21], [24] introduced the authorization mechanism into PKEET, where the users can specify an entity to perform the equality test and any unauthorized entity is unable to get correct test results. Ma et al. [22], [25] and Ma et al. [26] further designed flexible authorization mechanisms to satisfy various privacy requirements, for example, Alice can specify which ciphertexts can be compared by the entity.

In terms of basic constructions of PKEET, Zhang et al. [5], Zeng et al. [6] and Lee et al. [7] proposed the constructions in the standard model; Lee et al. [11] showed the generic constructions; Roy et al. [8] and Duong et al. [9] gave the constructions against quantum adversaries.

To decrease the workload of public key certificate distribution, Identity-Based Encryption with Equality Test (IBEET) schemes [27], [28], [29], [30] were presented. The offline message recovery attack is an inherit attack in PKEET and IBEET. Roughly speaking, given a ciphertext, the insider can pick a guessing message, encrypts it and then tests whether the resulting ciphertext and the given ciphertext contain the same message. Since the size of the message space is polynomial, the insider can efficiently implement its attack and recover the message. In order to resist this type of attack, the two-tester setting [31], [32] and the authentication in encryption [10], [33], [34] were suggested.

For richer functionality, Susilo et al. [2] introduced the multi-ciphertext equality test where the equality test can be performed among n ciphertexts for n users; Xu et al. [35] and Zhao et al. [36] presented the verifiable functionality where the equality test results can be verified by the users; Yang et al. [37] proposed the revocable revocation functionality where the users can revoke the test right of the third parties; Ma et al. [38] gave the time-based authorization for forward security.

In summery, the above work does not consider adaptive corruption attack and tight security reduction.

Organization. This paper will be organized as follows. Section II reviews several basic notions. Section III introduces the definition of PKEET. Section IV presents our PKEET scheme. In Section VI, we conclude this work.

II. PRELIMINARIES

Notation. Table 2 presents symbols, abbreviations and their descriptions.

TABLE 2. Symbols, abbreviations and their descriptions.

Symbol	Description
$s \xleftarrow{\$} S$	s is picked at random from a finite set S
λ	Security parameter
PPT	Probabilistic Polynomial Time
OW	OneWayness
IND	Indistinguishability
CPA	Chosen Plaintext Attacks
CCA	Chosen Ciphertext Attacks

Definition 1 (Decisional Diffie-Hellman (DDH) Assumption): For any PPT adversary $\mathcal{A}$ the following advantage function is negligible in λ.

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda) = \big|&\Pr\left[\mathcal{A}(q, G, g, g^u, g^v, g^{uv}) = 1\right] \\ &- \Pr\left[\mathcal{A}(q, G, g, g^u, g^v, g^w) = 1\right]\big|,\end{aligned}$$

where $u, v, w \xleftarrow{\$} \mathbb{Z}_q$.

Definition 2 (One-way Hash Function): A one-way hash function H can be efficiently computed, but for any PPT adversary $\mathcal{A}$ the following advantage function is negligible in λ.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OW}}(\lambda) = \Pr\left[\mathcal{A}(y = \mathsf{H}(x)) = x\right]$$

where $x \xleftarrow{\$} \{0, 1\}^*$.

III. DEFINITION OF PKEET

A. FORMAL DEFINITION

We propose the syntax of PKEET.

Definition 3 (Syntax of PKEET): PKEET consists of six PPT algorithms:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: The setup algorithm takes as input a security parameter λ, and outputs a public parameter pp.
- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{pk}, \mathsf{sk})$: The key generation algorithm takes as input the public parameter pp, and outputs a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \to \mathsf{ct}$: The encryption algorithm takes as input a public key pk and a message m, and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{m}$: The decryption algorithm takes as input a secret key sk and a ciphertext ct, and outputs the message m.
- $\mathsf{Aut}(\mathsf{sk}) \to \mathsf{td}$: The authorization algorithm takes as input a secret key sk, and outputs a trapdoor td.
- $\mathsf{Test}(\mathsf{td}, \mathsf{td}', \mathsf{ct}, \mathsf{ct}') \to 0/1$: The test algorithm takes as input two trapdoors $\mathsf{td}, \mathsf{td}'$ and two ciphertexts $\mathsf{ct}, \mathsf{ct}'$, and outputs 1 or 0.

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

1) For $\forall\lambda \in \mathbb{Z}^+$ and $\forall \mathsf{m} \in \mathcal{M}$, it holds that

$$\Pr\left[\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \,\middle|\, \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}) \end{array}\right] = 1.$$

2) For $\forall\lambda \in \mathbb{Z}^+$, $\forall \mathsf{m}, \mathsf{m}' \in \mathcal{M}$, if $\mathsf{m} = \mathsf{m}'$, it holds that

$$\Pr\left[\mathsf{Test}\begin{pmatrix} \mathsf{td}, \\ \mathsf{td}', \\ \mathsf{ct}, \\ \mathsf{ct}' \end{pmatrix} = 1 \,\middle|\, \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ (\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \\ \mathsf{ct}' \leftarrow \mathsf{Enc}(\mathsf{pk}', \mathsf{m}') \\ \mathsf{td} \leftarrow \mathsf{Aut}(\mathsf{sk}) \\ \mathsf{td}' \leftarrow \mathsf{Aut}(\mathsf{sk}') \end{array}\right] = 1.$$

3) For $\forall\lambda \in \mathbb{Z}^+$, $\forall \mathsf{m}, \mathsf{m}' \in \mathcal{M}$, if $\mathsf{m} \neq \mathsf{m}'$, it holds that

$$\Pr\left[\mathsf{Test}\begin{pmatrix} \mathsf{td}, \\ \mathsf{td}', \\ \mathsf{ct}, \\ \mathsf{ct}' \end{pmatrix} = 1 \,\middle|\, \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ (\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \\ \mathsf{ct}' \leftarrow \mathsf{Enc}(\mathsf{pk}', \mathsf{m}') \\ \mathsf{td} \leftarrow \mathsf{Aut}(\mathsf{sk}) \\ \mathsf{td}' \leftarrow \mathsf{Aut}(\mathsf{sk}') \end{array}\right]$$

is negligible in λ.

B. SECURITY NOTIONS

We consider the following adversaries:

- Type-I adversary who can obtain trapdoors issued by users, namely, testers;
- Type-II adversary who cannot obtain trapdoors, namely, dishonest users;

We first define ADaptive One-Way under Chosen Plaintext Attacks (AD-OW-CPA) security against Type-I adversary.

Definition 4 (AD-OW-CPA *security against* Type-I *adversary*): The game played between a challenger $\mathcal{C}$ and an Type-I adversary $\mathcal{A}_1$ is defined as follow.

- $\mathcal{C}$ runs $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ once and then $\mathsf{KeyGen}(\mathsf{pp})$ μ times to generate μ key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for $i \in [\mu]$, and sends $\mathsf{pp}, \mathsf{pk}_1, \mathsf{pk}_2, \ldots, \mathsf{pk}_\mu$ to $\mathcal{A}_1$.
- $\mathcal{O}_{\mathsf{sk}}$. $\mathcal{A}_1$ submits an index $i \in [\mu]$. $\mathcal{C}$ returns the secret key sk_i, and updates $Q_{\mathsf{sk}} = Q_{\mathsf{sk}} \cup \{i\}$.
- $\mathcal{O}_{\mathsf{td}}$. $\mathcal{A}_1$ submits an index $i \in [\mu]$. $\mathcal{C}$ runs $\mathsf{td}_i \leftarrow \mathsf{Aut}(\mathsf{sk}_i)$, returns the trapdoor td_i.
- $\mathcal{O}_{\mathsf{ct}}$. $\mathcal{A}_1$ submits an index $i^* \in [\mu]$. $\mathcal{C}$ randomly picks a message $\mathsf{m} \xleftarrow{\$} \mathcal{M}$ and then runs $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}_{i^*}, \mathsf{m})$, returns the challenge ciphertext ct^*, and updates $Q_{\mathsf{E}} = Q_{\mathsf{E}} \cup \{i^*\}$. We note that this oracle can be queried once.
- Finally, $\mathcal{A}_1$ outputs a message m', and wins the game if $\mathsf{m}' = \mathsf{m}$.

We say that a PKEET scheme is AD-OW-CPA secure against Type-I adversary $\mathcal{A}_1$ if for $Q_{\mathsf{sk}} \cap Q_{\mathsf{E}} = \emptyset$, the advantage function is negligible in λ, μ, namely,

$$\mathsf{Adv}^{\text{AD-OW-CPA}}_{\text{PKEET},\mathcal{A}_1}(\lambda, \mu) \leq \mathsf{negl}(\lambda).$$

Next we define ADaptive INDistinguishability under Chosen Plaintext Attacks (AD-IND-CPA) security against Type-II adversary.

Definition 5 (AD-IND-CPA *security against* Type-II *adversary*): The game played between a challenger $\mathcal{C}$ and an Type-II adversary $\mathcal{A}_2$ is defined as follow.

- $\mathcal{C}$ runs $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ once and then $\mathsf{KeyGen}(\mathsf{pp})$ μ times to generate μ key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for $i \in [\mu]$, and sends $\mathsf{pp}, \mathsf{pk}_1, \mathsf{pk}_2, \ldots, \mathsf{pk}_\mu$ to $\mathcal{A}_2$. It tosses a coin $\beta \xleftarrow{\$} \{0, 1\}$.
- $\mathcal{O}_{\mathsf{sk}}$. $\mathcal{A}_2$ submits an index $i \in [\mu]$. $\mathcal{C}$ returns the secret key sk_i, and updates $Q_{\mathsf{sk}} = Q_{\mathsf{sk}} \cup \{i\}$.
- $\mathcal{O}_{\mathsf{ct}}$. $\mathcal{A}_2$ submits an index $i^* \in [\mu]$ and two messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$. $\mathcal{C}$ runs $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}_{i^*}, \mathsf{m}_\beta)$, returns the challenge ciphertext ct^*, and updates $Q_{\mathsf{E}} = Q_{\mathsf{E}} \cup \{i^*\}$. We note that this oracle can be queried once.
- Finally, $\mathcal{A}_2$ outputs a bit β', and wins the game if $\beta' \in \beta$.

We say that a PKEET scheme is AD-IND-CPA secure against Type-II adversary $\mathcal{A}_2$ if for $Q_{\mathsf{sk}} \cap Q_{\mathsf{E}} = \emptyset$, the advantage function is negligible in λ, μ, namely,

$$\mathsf{Adv}^{\text{AD-IND-CPA}}_{\text{PKEET},\mathcal{A}_2}(\lambda, \mu) \leq 1/2 + \mathsf{negl}(\lambda).$$

IV. THE PROPOSED PKEET SCHEME

We present our PKEET scheme.

- $\mathsf{Setup}(1^\lambda)$: It takes as input a security parameter λ, and generates a public parameter pp as follows.
  1) Generate a group description $\mathbb{G} = (q, G, g)$.
  2) Sample $a \xleftarrow{\$} \mathbb{Z}_q$ and compute g^a.
  3) Pick a *one-way* hash function H: $\{0, 1\}^* \rightarrow G$.

  Output a public parameter pp.
  $$\mathsf{pp} =: (q, G, g, g^a, \mathsf{H}).$$
- $\mathsf{KeyGen}(\mathsf{pp})$: It takes as input the public parameter pp, samples $x_1, x_2, y_1, y_1 \xleftarrow{\$} \mathbb{Z}_q$, computes
  $$\mathsf{pk} := (g^{x_1+ax_2}, g^{y_1+ay_2}),$$
  $$\mathsf{sk} := (x_1, x_2, y_1, y_2).$$
  and outputs a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$: It takes as input a public key pk and a message m, samples $r \xleftarrow{\$} \mathbb{Z}_q$, computes
  $$U := g^r, V := g^{ar},$$
  $$W := g^{(x_1+ax_2)r} \cdot \mathsf{H}(\mathsf{m}),$$
  $$Z := g^{(y_1+ay_2)r} \cdot \mathsf{m},$$
  and outputs a ciphertext ct
  $$\mathsf{ct} := (U, V, W, Z).$$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: It takes as input a secret key sk and a ciphertext ct, and outputs the message
  $$\mathsf{m}' := Z/(U^{y_1} \cdot V^{y_2}).$$
- $\mathsf{Aut}(\mathsf{sk})$: It takes as input a secret key sk, and outputs a trapdoor td
  $$\mathsf{td} := (x_1, x_2).$$
- $\mathsf{Test}(\mathsf{td}, \mathsf{td}', \mathsf{ct}, \mathsf{ct}')$: It takes as input two trapdoors $\mathsf{td} = (x_1, x_2)$, $\mathsf{td}' = (x_1', x_2')$ and two ciphertexts $\mathsf{ct} = (U, V, W, Z)$, $\mathsf{ct}' = (U', V', W', Z')$, and checks whether the equation holds
  $$W/(U^{x_1} \cdot V^{x_2}) = W'/(U'^{x_1'} \cdot V'^{x_2'}).$$
  If so, it outputs 1; otherwise, it outputs 0.

The correctness is demonstrated as follows:

1) As for the first condition, we have that
$$\begin{aligned} &Z/(U^{y_1} \cdot V^{y_2}) \\ &\quad = (g^{(y_1+ay_2)r} \cdot \mathsf{m})/((g^r)^{y_1} \cdot (g^{ar})^{y_2}) \\ &\quad = \mathsf{m} \end{aligned}$$
2) As for the second condition, if $\mathsf{m} = \mathsf{m}'$, we have that
$$\begin{aligned} &W/(U^{x_1} \cdot V^{x_2}) \\ &\quad = (g^{(x_1+ax_2)r} \cdot \mathsf{H}(\mathsf{m}))/((g^r)^{x_1} \cdot (g^{ar})^{x_2}) \\ &\quad = \mathsf{H}(\mathsf{m}) \\ &W'/(U'^{x_1'} \cdot V'^{x_2'}) \\ &\quad = (g^{(x_1'+ax_2')r'} \cdot \mathsf{H}(\mathsf{m}'))/((g^{r'})^{x_1'} \cdot (g^{ar'})^{x_2'}) \\ &\quad = \mathsf{H}(\mathsf{m}') \end{aligned}$$
The equation holds as $\mathsf{H}(\mathsf{m}) = \mathsf{H}(\mathsf{m}')$.
3) According to 2), the third condition must hold.

V. SECURITY ANALYSIS

Theorem 1: For any PPT Type-I adversary $\mathcal{A}_1$ who makes at most q_{sk} and q_{td} queries to O_{sk} and O_{td} respectively and one query to O_{ct}, there exist $\mathcal{B}$ such that

$$\mathsf{Adv}^{\text{AD-OW-CPA}}_{\text{PKEET},\mathcal{A}_1}(1^\lambda, \mu) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{OW}}_{\mathsf{H}}(\lambda) + 1/q.$$

Proof: We define the advantage function of any PPT adversary $\mathcal{A}_1$ in Game_x as

$$\mathsf{Adv}^{\mathsf{Game}_x}_{\mathcal{A}_1}(\lambda).$$

- Game_0: is the real game. We thus have that
$$\mathsf{Adv}^{\text{AD-OW-CPA}}_{\text{PKEET},\mathcal{A}_1}(1^\lambda, \mu) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}_1}(\lambda).$$
- Game_1: is identical to Game_0 except that the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is converted into the following form:
$$U := g^r, V := \boxed{g^{r'}},$$
$$W := \boxed{g^{x_1^{(i^*)}r + x_2^{(i^*)}r'}} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := \boxed{g^{y_1^{(i^*)}r + y_2^{(i^*)}r'}} \cdot \mathsf{m}.$$

Lemma 1: For any PPT adversary $\mathcal{A}_1$,

$$\left|\mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}_1}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}_1}(\lambda)\right| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{B}}(\lambda).$$

Proof: We describe the simulation as below.

- Given an instance (q, G, g, g^u, g^v, T) of the DDH problem where either $T = g^{uv}$ or $T = g^w$, $\mathcal{B}$ selects a one-way hash function $\mathsf{H} : \{0, 1\}^* \to G$, and sets
$$\mathsf{pp} =: (q, G, g, g^u, \mathsf{H}),$$
where we implicitly define $g^a = g^u$. For $i \in [\mu]$, $\mathcal{B}$ picks $x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)} \xleftarrow{\$} \mathbb{Z}_q$ and generates
$$\mathsf{pk}_i := (g^{x_1^{(i)} + ux_2^{(i)}}, g^{y_1^{(i)} + uy_2^{(i)}}),$$
$$\mathsf{sk}_i := (x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)}).$$
Send $\mathsf{pp}, \mathsf{pk}_1, \ldots, \mathsf{pk}_\mu$ to $\mathcal{A}_1$.
- O_{sk}: Given an index i, $\mathcal{B}$ returns $\mathsf{sk}_i = (x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)})$, and updates $Q_{\mathsf{sk}} = Q_{\mathsf{sk}} \cup \{i\}$, where Q_{sk} is an initially empty set.
- O_{td}: Given an index i, $\mathcal{B}$ returns $\mathsf{td}_i = (x_1^{(i)}, x_2^{(i)})$ and updates $Q_{\mathsf{td}} = Q_{\mathsf{td}} \cup \{i\}$, where Q_{td} is an initially empty set.
- O_{ct}: Given an index i^*, $\mathcal{B}$ picks $\mathsf{m} \xleftarrow{\$} \mathcal{M}$, forms challenge ciphertext as
$$U := g^v, V := T,$$
$$W := U^{x_1^{(i^*)}} \cdot V^{x_2^{(i^*)}} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := U^{y_1^{(i^*)}} \cdot V^{y_2^{(i^*)}} \cdot \mathsf{m},$$
returns $\mathsf{ct}^* = (U, V, W, Z)$, and updates $Q_{\mathsf{E}} = Q_{\mathsf{E}} \cup \{i^*\}$, where Q_{E} is an initially empty set.
- Finally, $\mathcal{A}_1$ outputs a message m', and wins the game if $\mathsf{m}' = \mathsf{m}$.

Analysis. We claim that if $T = g^{uv}$, the challenge ciphertext ct^* is properly distributed as the challenge ciphertext in Game_0. To see this, ct^* is formed as

$$U := g^v, V := g^{uv},$$
$$W := (g^v)^{x_1^{(i^*)}} \cdot (g^{uv})^{x_2^{(i^*)}} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := (g^v)^{y_1^{(i^*)}} \cdot (g^{uv})^{y_2^{(i^*)}} \cdot \mathsf{m},$$

that is,

$$U := g^v, V := g^{uv},$$
$$W := g^{(x_1^{(i^*)} + ux_2^{(i^*)})v} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := g^{(y_1^{(i^*)} + uy_2^{(i^*)})v} \cdot \mathsf{m}.$$

Note that we implicitly define $r = v$. Otherwise, we have that $T = g^w$. The challenge ciphertext ct^* is properly distributed in Game_1. To see this, ct^* is formed as

$$U := g^v, V := g^w,$$
$$W := (g^v)^{x_1^{(i^*)}} \cdot (g^w)^{x_2^{(i^*)}} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := (g^v)^{y_1^{(i^*)}} \cdot (g^w)^{y_2^{(i^*)}} \cdot \mathsf{m},$$

that is,

$$U := g^v, V := g^w,$$
$$W := g^{x_1^{(i^*)}v + x_2^{(i^*)}w} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := g^{y_1^{(i^*)}v + y_2^{(i^*)}w} \cdot \mathsf{m}.$$

Note that we implicitly define $r' = w$. □

- Game_2: is identical to Game_1 except that the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is converted into the following form:
$$U := g^r, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r + x_2^{(i^*)}r'} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := g^{y_1^{(i^*)}r + y_2^{(i^*)}r'} \cdot \boxed{\mathsf{m}_R}.$$

Lemma 2: For any PPT adversary $\mathcal{A}_1$,

$$\left|\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}_1}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}_2}(\lambda)\right| = 1/q.$$

Proof: Observe that in Game_1, the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is formed as

$$U := g^r, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r + x_2^{(i^*)}r'} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := g^{y_1^{(i^*)}r + y_2^{(i^*)}r'} \cdot \mathsf{m},$$

We argue that Z is exactly a perfect one-time pad, and thus in Game_2, we can replace the message m with a random message m_R but with a small error. It suffices to show that

$$y_1^{(i^*)}r + y_2^{(i^*)}r' \tag{1}$$

is uniform over $\mathbb{Z}_q$. Considering $\mathcal{A}_1$ is given

$$y_1^{(i^*)} + ay_2^{(i^*)} \tag{2}$$

from the public key pk_{i^*}. Using (1) and (2), we have that

$$\begin{pmatrix} y_1^{(i^*)} + ay_2^{(i^*)} \\ y_1^{(i^*)}r + y_2^{(i^*)}r' \end{pmatrix} = \begin{pmatrix} 1 & a \\ r & r' \end{pmatrix} \cdot \begin{pmatrix} y_1^{(i^*)} \\ y_2^{(i^*)} \end{pmatrix}.$$

Since the determinant of the above matrix is not equal to 0, the solution is unique. Hence when $y_1^{(i^*)}$ and $y_2^{(i^*)}$ are picked at random, $y_1^{(i^*)}r + y_2^{(i^*)}r'$ is uniform over $\mathbb{Z}_q$. Therefore, we can replace the message m with a random message m_R, namely, the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is formed as

$$U := g^r, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r + x_2^{(i^*)}r'} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := g^{y_1^{(i^*)}r + y_2^{(i^*)}r'} \cdot \mathsf{m}_R,$$

□

In Game_2, only W contain the information about the message m. We argue that the adversary $\mathcal{A}_1$ can recover the message with a negligible probability.

Lemma 3: For any PPT adversary $\mathcal{A}_1$,

$$\mathsf{Adv}_{\mathcal{A}_1}^{\mathsf{Game}_2}(\lambda) \le \mathsf{Adv}_{\mathsf{H}}^{\mathsf{OW}}(\lambda).$$

Proof: Observe that in Game_2, the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is formed as

$$U := g^r, V := g^{r'},$$
$$W := g^{x_1 r + x_2 r'} \cdot \mathsf{H}(\mathsf{m}),$$
$$Z := g^{y_1 r + y_2 r'} \cdot \mathsf{m}_R.$$

We note that $\mathcal{A}_1$ can obtain all trapdoors, thus it is easy for $\mathcal{A}_1$ to get $\mathsf{H}(\mathsf{m})$. But if $\mathcal{A}_1$ can find out the message from $\mathsf{H}(\mathsf{m})$, there must be an efficient algorithm breaking the one-wayness of the hash function H. □

This completes the proof. □

Theorem 2: For any PPT Type-II adversary $\mathcal{A}_2$ who makes at most q_{sk} queries to O_{sk} and one query to O_{ct}, there exist $\mathcal{B}$ such that

$$\mathsf{Adv}_{\mathrm{PKEET},\mathcal{A}_2}^{\text{AD-IND-CPA}}(1^\lambda, \mu) \le \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}}(\lambda) + 2/q + 1/2.$$

Proof: We define the advantage function of any PPT adversary $\mathcal{A}_2$ in Game_x as

$$\mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_x}(\lambda).$$

– Game_0: is the real game. We have that

$$\mathsf{Adv}_{\mathrm{PKEET},\mathcal{A}_2}^{\text{AD-IND-CPA}}(1^\lambda, \mu) = \mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_0}(\lambda).$$

– Game_1: is identical to Game_0 except that the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is converted into the following form:

$$U := g^r, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r + x_2^{(i^*)}r'} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := g^{y_1^{(i^*)}r + y_2^{(i^*)}r'} \cdot \mathsf{m}_\beta.$$

Lemma 4: For any PPT adversary $\mathcal{A}_2$,

$$\left|\mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_0}(\lambda) - \mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_1}(\lambda)\right| \le \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}}(\lambda).$$

Proof: We describe the simulation as below.

- Given an instance (q, G, g, g^u, g^v, T) of the DDH problem where either $T = g^{uv}$ or $T = g^w$, $\mathcal{B}$ selects a one-way hash function $\mathsf{H} : \{0, 1\}^* \to G$, and sets

$$\mathsf{pp} =: (q, G, g, g^u, \mathsf{H}),$$

where we implicitly define $g^a = g^u$. For $i \in [\mu]$, $\mathcal{B}$ picks $x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)} \xleftarrow{\$} \mathbb{Z}_q$ and generates

$$\mathsf{pk}_i := (g^{x_1^{(i)} + ux_2^{(i)}}, g^{y_1^{(i)} + uy_2^{(i)}}),$$
$$\mathsf{sk}_i := (x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)}).$$

It sends $\mathsf{pp}, \mathsf{pk}_1, \ldots, \mathsf{pk}_\mu$ to $\mathcal{A}_2$. Tosse a coin $\beta \xleftarrow{\$} \{0, 1\}$.
- O_{sk}: Given an index i, $\mathcal{B}$ returns $\mathsf{sk}_i = (x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)})$, and updates $Q_{\mathsf{sk}} = Q_{\mathsf{sk}} \cup \{i\}$, where Q_{sk} is an initially empty set.
- O_{ct}: Given an index i^* and two messages $\mathsf{m}_0, \mathsf{m}_1$, $\mathcal{B}$ forms the challenge ciphertext as

$$U := g^v, V := T',$$
$$W := U^{x_1^{(i^*)}} \cdot V^{x_2^{(i^*)}} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := U^{y_1^{(i^*)}} \cdot V^{y_2^{(i^*)}} \cdot \mathsf{m}_\beta,$$

returns $\mathsf{ct}^* = (U, V, W, Z)$, and updates $Q_{\mathsf{E}} = Q_{\mathsf{E}} \cup \{i^*\}$, where Q_{E} is an initially empty sets.
- Finally, $\mathcal{A}_2$ outputs a bit β', and wins the game if $\beta' = \beta$.

Analysis. We claim that if $T = g^{uv}$, the challenge ciphertext ct^* is properly distributed as the challenge ciphertext in Game_0. To see this, ct^* is formed as

$$U := g^v, V := g^{uv},$$
$$W := (g^v)^{x_1^{(i^*)}} \cdot (g^{uv})^{x_2^{(i^*)}} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := (g^v)^{y_1^{(i^*)}} \cdot (g^{uv})^{y_2^{(i^*)}} \cdot \mathsf{m}_\beta,$$

that is,

$$U := g^v, V := g^{uv},$$
$$W := g^{(x_1^{(i^*)} + ux_2^{(i^*)})v} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := g^{(y_1^{(i^*)} + uy_2^{(i^*)})v} \cdot \mathsf{m}_\beta.$$

Note that we implicitly define $r = v$. Otherwise, we have that $T = g^w$. The challenge ciphertext ct^* is properly distributed in Game_1. To see this, ct^* is formed as

$$U := g^v, V := g^w,$$
$$W := (g^v)^{x_1^{(i^*)}} \cdot (g^w)^{x_2^{(i^*)}} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := (g^v)^{y_1^{(i^*)}} \cdot (g^w)^{y_2^{(i^*)}} \cdot \mathsf{m}_\beta,$$

that is,

$$U := g^{v}, V := g^{w},$$
$$W := g^{x_1^{(i^*)}v+x_2^{(i^*)}w} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := g^{y_1^{(i^*)}v+y_2^{(i^*)}w} \cdot \mathsf{m}_\beta.$$

Note that we implicitly define $r' = w$. □

– Game_2: is identical to Game_1 except that the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is converted into the following form:

$$U := g^{r}, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r+x_2^{(i^*)}r'} \cdot \boxed{\mathsf{H}(\mathsf{m}_{R'})},$$
$$Z := g^{y_1^{(i^*)}r+y_2^{(i^*)}r'} \cdot \boxed{\mathsf{m}_R}.$$

Lemma 5: For any PPT adversary $\mathcal{A}_2$,

$$\left|\mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_2}(\lambda)\right| = 2/q.$$

Proof: Observe in Game_1, the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is formed as

$$U := g^{r}, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r+x_2^{(i^*)}r'} \cdot \mathsf{H}(\mathsf{m}_\beta),$$
$$Z := g^{y_1^{(i^*)}r+y_2^{(i^*)}r'} \cdot \mathsf{m}_\beta.$$

Followed by the proof of Lemma 2, it is not difficult to get that Z is exactly a perfect one-time pad. We now argue that W is exactly a perfect one-time pad as well, and thus in Game_2, we can replace the message m_β with a random message $m_{R'}$ but with a small error. It suffices to show that

$$x_1^{(i^*)}r + x_2^{(i^*)}r' \tag{3}$$

is uniform over $\mathbb{Z}_q$. Considering $\mathcal{A}_2$ is given

$$x_1^{(i^*)} + ax_2^{(i^*)} \tag{4}$$

from the public key pk_{i^*}. Using (3) and (4), we have that

$$\begin{pmatrix} x_1^{(i^*)} + ax_2^{(i^*)} \\ x_1^{(i^*)}r + x_2^{(i^*)}r' \end{pmatrix} = \begin{pmatrix} 1 & a \\ r & r' \end{pmatrix} \cdot \begin{pmatrix} x_1^{(i^*)} \\ x_2^{(i^*)} \end{pmatrix}$$

Since the determinant of the above matrix is not equal to 0, the solution is unique. Hence when $x_1^{(i^*)}$, $x_2^{(i^*)}$ are picked at random, $x_1^{(i^*)}v + x_2^{(i^*)}w$ is uniform over $\mathbb{Z}_q$. Therefore, we can replace the message m_β with two random message m_R, $\mathsf{m}_{R'}$, namely, the challenge ciphertext $\mathsf{ct}^* = (U, V, W, Z)$ is formed as

$$U := g^{r}, V := g^{r'},$$
$$W := g^{x_1^{(i^*)}r+x_2^{(i^*)}r'} \cdot \mathsf{H}(\mathsf{m}_{R'}),$$
$$Z := g^{y_1^{(i^*)}r+y_2^{(i^*)}r'} \cdot \mathsf{m}_R.$$

□

In Game_2, there is no information about the message m_β. Therefore, the adversary $\mathcal{A}_2$ can guess β with probability $1/2$, namely,

$$\mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{Game}_2}(\lambda) = 1/2.$$

This completes the proof. □

VI. CONCLUSION

In this paper, we discussed real corruption attack and tight security reduction for PKEET. Firstly, we pointed out that in order to derive reasonable security against the real attackers, the security models should reflect the real attacks as closely as possible. Thus, we have to capture the real corruption attack in the security model and allow the adversary to adaptively corrupt users. Secondly, we argued that tight security reduction is meaningful to the implementation of the scheme. However, many PKEET schemes suffer from a security loss of μ in proving adaptive security. Finally, we presented a tightly secure PKEET scheme in setting with adaptive corruptions and showed our techniques.

For the future work, we will improve our security model. Concretely, we consider a strong security model in which the adversary can request multiple ciphertexts to attack, which can further narrow the gap between security model of PKEET and the real attacks, and derive concrete security guarantees against the real attackers. We note that in the real word, there are always many ciphertexts in the system. We note also that our current techniques cannot prove tight security in the multi-ciphertext setting, since the entropy provided by public keys is insufficient to hide many messages. This motivates us to study new proof techniques.

REFERENCES

[1] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Topics in Cryptology-CT-RSA*. Berlin, Germany: Springer, 2010, pp. 119–131.

[2] W. Susilo, F. Guo, Z. Zhao, and G. Wu, "PKE-MET: Public-key encryption with multi-ciphertext equality test in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 1476–1488, Apr. 2022.

[3] Y. Wang, Q. Huang, H. Li, M. Xiao, S. Ma, and W. Susilo, "Private set intersection with authorization over outsourced encrypted datasets," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4050–4062, 2021.

[4] S. Ma, P. Zhou, Q. Huang, and J. Wang, "MTER: An efficient multi-user threshold equality retrieval for double auction in smart grid market," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4515–4529, 2023.

[5] K. Zhang, J. Chen, H. T. Lee, H. Qian, and H. Wang, "Efficient public key encryption with equality test in the standard model," *Theor. Comput. Sci.*, vol. 755, pp. 65–80, Jan. 2019.

[6] M. Zeng, J. Chen, K. Zhang, and H. Qian, "Public key encryption with equality test via hash proof system," *Theor. Comput. Sci.*, vol. 795, pp. 20–35, Nov. 2019.

[7] H. T. Lee, S. Ling, J. H. Seo, and H. Wang, "Public key encryption with equality test from generic assumptions in the random Oracle model," *Inf. Sci.*, vol. 500, pp. 15–33, Oct. 2019.

[8] P. S. Roy, D. H. Duong, W. Susilo, A. Sipasseuth, K. Fukushima, and S. Kiyomoto, "Lattice-based public-key encryption with equality test supporting flexible authorization in standard model," *Theor. Comput. Sci.*, vol. 929, pp. 124–139, Sep. 2022.

[9] D. H. Duong, P. S. Roy, W. Susilo, K. Fukushima, S. Kiyomoto, and A. Sipasseuth, "Chosen-ciphertext lattice-based public key encryption with equality test in standard model," *Theor. Comput. Sci.*, vol. 905, pp. 31–53, Feb. 2022.

[10] Y. Ling, S. Ma, Q. Huang, X. Li, and Y. Ling, "Group public key encryption with equality test against offline message recovery attack," *Inf. Sci.*, vol. 510, pp. 16–32, Feb. 2020.
[11] H. T. Lee, S. Ling, J. H. Seo, H. Wang, and T.-Y. Youn, "Public key encryption with equality test in the standard model," *Inf. Sci.*, vol. 516, pp. 89–108, Apr. 2020.
[12] Y. Lu, R. Zhang, and D. Lin, "Stronger security model for public-key encryption with equality test," in *Proc. Int. Conf. Pairing-Based Cryptogr.*, 2012, pp. 65–82.
[13] Y. Lee, D. H. Lee, and J. H. Park, "Tightly CCA-secure encryption scheme in a multi-user setting with corruptions," *Des., Codes Cryptogr.*, vol. 88, no. 11, pp. 2433–2452, Nov. 2020.
[14] S. Han, S. Liu, and D. Gu, "Key encapsulation mechanism with tight enhanced security in the multi-user setting: Impossibility result and optimal tightness," in *Proc. ASIACRYPT*, 2021, pp. 483–513.
[15] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1998, pp. 127–144.
[16] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, Feb. 2006.
[17] G. Ateniese, K. Benson, and S. Hohenberger, "Key-private proxy re-encryption," in *Proc. Cryptographers' Track RSA Conf.* Cham, Switzerland: Springer, 2009, pp. 279–294.
[18] A. Cohen, "What about Bob? The inadequacy of CPA security for proxy reencryption," in *Proc. IACR Int. Workshop Public Key Cryptogr.* Cham, Switzerland: Springer, 2019, pp. 287–316.
[19] G. Fuchsbauer, C. Kamath, K. Klein, and K. Pietrzak, "Adaptively secure proxy re-encryption," in *Proc. IACR Int. Workshop Public Key Cryptogr.* Cham, Switzerland: Springer, Apr. 2019, pp. 317–346.
[20] J. Chen and H. Wee, "Fully, (almost) tightly secure IBE and dual system groups," in *Proc. CRYPTO*, 2013, pp. 435–460.
[21] Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," *Secur. Commun. Netw.*, vol. 5, no. 12, pp. 1351–1362, Dec. 2012.
[22] S. Ma, M. Zhang, Q. Huang, and B. Yang, "Public key encryption with delegated equality test in a multi-user setting," *Comput. J.*, vol. 58, no. 4, pp. 986–1002, Apr. 2015.
[23] D. Boneh, G. Di. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 2004, pp. 506–522.
[24] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authorization," in *Proc. 16th Aust. Conf. Inf. Security Privacy*, 2011, pp. 389–406.
[25] S. Ma, Q. Huang, M. Zhang, and B. Yang, "Efficient public key encryption with equality test supporting flexible authorization," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 458–470, Mar. 2015.
[26] K. Huang, R. Tso, Y.-C. Chen, S. M. M. Rahman, A. Almogren, and A. Alamri, "PKE-AET: Public key encryption with authorized equality test," *Comput. J.*, vol. 58, no. 10, pp. 2686–2697, Oct. 2015.
[27] S. Ma, "Identity-based encryption with outsourced equality test in cloud computing," *Inf. Sci.*, vol. 328, pp. 389–402, Jan. 2016.
[28] X.-J. Lin, Q. Wang, L. Sun, and H. Qu, "Identity-based encryption with equality test and datestamp-based authorization mechanism," *Theor. Comput. Sci.*, vol. 861, pp. 117–132, Mar. 2021.
[29] L. Wu, Y. Zhang, K.-K.-R. Choo, and D. He, "Efficient and secure identity-based encryption scheme with equality test in cloud computing," *Future Gener. Comput. Syst.*, vol. 73, pp. 22–31, Aug. 2017.
[30] L. Wu, Y. Zhang, K. R. Choo, and D. He, "Efficient identity-based encryption scheme with equality test in smart city," *IEEE Trans. Sustain. Comput.*, vol. 3, no. 1, pp. 44–55, Jan. 2018.
[31] Q. Tang, "Public key encryption schemes supporting equality test with authorization of different granularity," *Int. J. Appl. Cryptogr.*, vol. 2, no. 4, pp. 304–321, 2012.
[32] T. Wu, S. Ma, Y. Mu, and S. Zeng, "ID-based encryption with equality test against insider attack," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Auckland, New Zealand, Jul. 2017, pp. 168–183.
[33] Y. Ling, S. Ma, Q. Huang, R. Xiang, and X. Li, "Group ID-based encryption with equality test," in *Proc. Australas. Conf. Inf. Secur. Privacy (ACISP)*, Jul. 2019, pp. 39–57.
[34] Y. Ling, S. Ma, Q. Huang, X. Li, Y. Zhong, and Y. Ling, "Efficient group ID-based encryption with equality test against insider attack," *Comput. J.*, vol. 64, no. 1, pp. 661–674, Nov. 2019.
[35] Y. Xu, M. Wang, H. Zhong, J. Cui, L. Liu, and V. N. L. Franqueira, "Verifiable public key encryption scheme with equality test in 5G networks," *IEEE Access*, vol. 5, pp. 12702–12713, 2017.
[36] Z. Zhao, W. Susilo, B. Wang, and K. Zeng, "Public-key encryption with tester verifiable equality test for cloud computing," *IEEE Trans. Cloud Comput.*, vol. 11, no. 4, pp. 3396–3406, Jun. 2023.
[37] T. Yang, S. Ma, J. Du, C. Jiang, and Q. Huang, "Revocable public key encryption with equality test without pairing in cloud storage," *Comput. J.*, vol. 67, no. 2, pp. 642–657, Feb. 2024.
[38] S. Ma, Z. Ye, Q. Huang, and C. Jiang, "Controllable forward secure identity-based encryption with equality test in privacy-preserving text similarity analysis," *Inf. Sci.*, vol. 66, Mar. 2024, Art. no. 20099.

YUNHAO LING is currently pursuing the Ph.D. degree with the School of Software Engineering, East China Normal University, Shanghai, China. His research interests include public key encryption, provable security, and security in cloud computing.