New RSA-Based Public Key Encryption with Authorized Equality Test

ChanHyeok Park, Seongbong Choi, Yongseok Son, Jeongyeup Paek, Sungrae Cho, and Hyung Tae Lee
School of Computer Science and Engineering
Chung-Ang University
Seoul, Republic of Korea
{jchcaksj, welq2st, sysganda, jpaek, srcho, hyungtaelee}@cau.ac.kr

***Abstract*—This article proposes a new RSA-based public key encryption scheme with authorized equality test (PKE-AET). Our construction satisfies the one-wayness against adaptive chosen ciphertext attacks for adversaries who may get a token for equality test under the RSA assumption in the random oracle model. When the adversary cannot get a token for test, it satisfies the indistinguishability against adaptive chosen ciphertext attacks under the same assumption. Furthermore, compared to other existing RSA-based PKE-AET constructions, our construction additionally enables each user to issue a token for a particular ciphertext as well as all ciphertexts of user, while achieving comparable efficiency.**

***Index Terms*—Public key encryption, authorized equality test, RSA assumption, random oracle model**

I. Introduction

Public key encryption with equality test (PKEET) is a special type of public key encryption scheme that enables to check equality between plaintexts contained in two ciphertexts regardless of the equality of underlying public keys. Since its concept with concrete instantiation was proposed by Yang, Tan, Huang, and Wong [1], various PKEET schemes have been suggested, due to its diverse applications, like secure email spam-filtering system and secure data management in the Internet of vehicles. Among them, almost all existing constructions were based on the hardness of the discrete logarithm (DL)-based cryptographic problems [2]–[6] or lattice-based cryptographic hard problems [7], [8].

On the other hand, there are only a few PKEET constructions in the RSA-based setting. In [9], a concrete RSA-based PKEET construction was proposed by Zhu, Xie, Ahmad, and Hasan Abdullah. However, their construction achieves the security against non-adaptive chosen ciphertext attacks (CCA1) only. We can also obtain RSA-based instantiations from generic constructions [10], [11] for PKEET by employing RSA-based public key encryption schemes, like RSA-OAEP [12]. Such instantiations can achieve the security against adaptive chosen ciphertext attacks (CCA2), but they allow each user to issue a token for all his/her ciphertexts only.

This article presents a variant of PKEET scheme, which we call a public key encryption with authorized equality test (PKE-AET), under the RSA setting. While aforementioned RSA-based constructions allow to issue a token for all ciphertexts of user only, our construction allows to issue a token for a particular ciphertext as well as all ciphertexts of user. Under assuming that the RSA assumption holds, our proposed scheme achieves the onewayness against CCA2 (OW-CCA2) for adversaries who may get tokens and the indistinguishability against CCA2 (IND-CCA2) for adversaries who cannot get, in the random oracle model. In order to demonstrate the efficiency of our construction, we provide implementation results of ours and other RSA-based PKEET constructions. The experimental results show that ours has comparable efficiency, while supporting additional functionality and/or enhancing the security, compared to existing results.

Outline of the Paper. The next section introduces definitions of PKE-AET and cryptographic assumptions to be utilized to analyze the correctness and security of our proposed construction. In Section III, we give a new RSA-based PKE-AET construction and discuss its security. Then, in Section IV, we provide comparison of our construction and other existing results, including implementation results of our construction under various parameter settings.

II. Preliminaries

We first introduce definitions for PKE-AET, including its system model, a formal description of the scheme, and security definitions. We then present several cryptographic assumptions which will be utilized to analyze the security of our construction.

This section starts with introducing several notations that will be utilized in the paper.

Notation. For an integer $n \geq 2$, $\mathbb{Z}_n$ denotes the set of integers between 0 and $n-1$, including them, and $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \,|\, \gcd(a,n) = 1\}$. We denote by $|S|$ the cardinality of a set S. The function $\phi(n)$ is the Euler totient function, which is defined as the cardinality of $\mathbb{Z}_n^*$, i.e., $\phi(n) = |\mathbb{Z}_n^*|$.

When A is an algorithm, $A \rightarrow a$ indicates that a is an output of A. If S is a set or a distribution, $s \overset{\$}{\leftarrow} S$ means that

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A4A5034130), and also by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). H. T. Lee is the corresponding author.

979-8-3503-3094-6/24/$31.00 ©2024 IEEE

s is selected uniformly and randomly from S. We say that a function $f : \mathbb{N} \rightarrow \mathbb{R}$ is negligible in λ if there is a sufficiently large λ such that $f(\lambda) \leq \frac{1}{p(\lambda)}$ for all positive polynomials $p(\cdot)$.

A. Public Key Encryption with Authorized Equality Test

Now, we first look into the system model of PKE-AET.

System Model for PKE-AET. The PKE-AET system is composed of two kinds of entities, users (including senders and receivers) and tester(s). A sender, who would like to send a receiver a plaintext securely, encrypts a plaintext and passes a ciphertexet to the receiver. Once he/she receives the ciphertext from the sender, he/she may decrypt it and/or store it at the cloud. Later, if needed, he/she may generate a token for equality tests on either a particular ciphertext only or all his/her ciphertexts, and sends a tester it. Then, the tester who has the token can execute tests on either a particular ciphertext or all ciphertexts of user who passes the right of equality test with respect to types of token.

We formally define a PKE-AET scheme below.

Definition 1 (Public Key Encryption with Authorized Equality Test): A public key encryption with authorized equality test (PKE-AET) is composed of 7 polynomial time algorithms below:

- $\mathsf{Setup}(\lambda)$: Given a security parameter λ as an input, it outputs a public parameter pp.
- $\mathsf{KeyGen}(pp)$: On input pp, it outputs a pair of public and secret keys (pk, sk) of a user.
- $\mathsf{Enc}(pk, M)$: Taking pk and a plaintext M as inputs, it outputs a ciphertext CT.
- $\mathsf{Dec}(sk, CT)$: Taking sk and CT as inputs, it outputs a plaintext M'.
- $\mathsf{Aut}_1(sk)$: On input sk, it returns a token tk_1 for all ciphertexts of the user corresponding to the input sk.
- $\mathsf{Aut}_2(sk, CT)$: Taking sk and CT as inputs, it returns a token $tk_{2,CT}$ for equality test on CT.
- $\mathsf{Test}(CT_A, tk_A, CT_B, tk_B)$: Given two pairs of ciphertext and token, (CT_A, tk_A), (CT_B, tk_B), output 1 denoting that CT_A and CT_B have the same plaintext or 0 denoting that they have different plaintexts.
  We remark that tk_A and tk_B can be any types of tokens, generated by Aut_1 or Aut_2 algorithms.

Correctness of PKE-AET. Now, we look at the correctness of PKE-AET constructions. It can be divided into two parts: (1) the correctness of returning a plaintext in the decryption algorithm and (2) the correctness of checking equality.

Definition 2 (Correctness of PKE-AET): A PKE-AET construction is correct if it satisfies that

1) For any security parameter λ and plaintext $M \in \mathcal{M}$,
$$\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, M)) \rightarrow M]$$
is negligible in λ where $\mathsf{Setup}(\lambda) \rightarrow pp$ and $\mathsf{KeyGen}(pp) \rightarrow (pk, sk)$.
2) For any security parameter λ and plaintexts $M_A, M_B \in \mathcal{M}$,
   - if $\mathsf{Dec}(sk_A, CT_A) = \mathsf{Dec}(sk_B, CT_B) \neq \perp$,
   $$\Pr[\mathsf{Test}(CT_A, tk_A, CT_B, tk_B) \rightarrow 1] = 1,$$
   - if $\mathsf{Dec}(sk_A, CT_A) \neq \mathsf{Dec}(sk_B, CT_B)$,
   $$\Pr[\mathsf{Test}(CT_A, tk_A, CT_B, tk_B) \rightarrow 1]$$
   is negligible in λ,

   where $\mathsf{Setup}(\lambda) \rightarrow pp$, $\mathsf{KeyGen}(pp) \rightarrow (pk_A, sk_A)$, $\mathsf{KeyGen}(pp) \rightarrow (pk_B, sk_B)$, $\mathsf{Enc}(pk_A, M_A) \rightarrow CT_A$, $\mathsf{Enc}(pk_B, M_B) \rightarrow CT_B$, tk_A is the output of $\mathsf{Aut}_1(sk_A)$ or $\mathsf{Aut}_2(sk_A, CT_A)$, and tk_B is the output of $\mathsf{Aut}_1(sk_B)$ or $\mathsf{Aut}_2(sk_B, CT_B)$.

Security Model for PKE-AET. In the PKE-AET system, we may classify adversaries into two types with respect to whether they can get a token for equality tests or not.

- Type-I adversary: This adversary can get a token for equality test on target user's ciphertext(s). Hence, it enables to determine a plaintext involved in the target ciphertext between two candidates. Therefore, this adversary attempts at extracting the plaintext in the target ciphertext.
- Type-II adversary: This adversary cannot get a token for equality test on target user's ciphertext(s). Therefore, this adversary atempts at guessing a plaintext correctly in the target ciphertext between two candidates.

By reflecting the above features, we provide formal definitions of PKE-AET. We first define the OW-CCA2 security against a Type-I probabilistic polynomial time (PPT) adversary.

Definition 3 (OW-CCA2 Security against Type-I Adversaries): A PKE-AET scheme is OW-CCA2 secure against Type-I adversaries if there is no PPT adversary $\mathcal{A}$ whose advantage in the experiment $\mathbf{Exp}^{\text{OW-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$, defined as below, is negligible in the security parameter λ:

$\mathbf{Exp}^{\text{OW-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$

$\mathsf{Setup}(\lambda) \rightarrow (pp)$
$\mathsf{KeyGen}(pp) \rightarrow (pk_t, sk_t)$
$\mathcal{A}^{\mathcal{O}^{\mathsf{KG}}, \mathcal{O}^{\mathsf{Aut}_1}, \mathcal{O}^{\mathsf{Aut}_2}, \mathcal{O}^{\mathsf{Dec}}}(pk_t) \rightarrow st$
$M \xleftarrow{\$} \mathcal{M}$
$\mathsf{Enc}(pk_t, M) \rightarrow CT^*$
$\mathcal{A}^{\mathcal{O}^{\mathsf{KG}}, \mathcal{O}^{\mathsf{Aut}_1}, \mathcal{O}^{\mathsf{Aut}_2}, \mathcal{O}^{\mathsf{Dec}}}(CT^*) \rightarrow M'$

In the above experiment, the oracles $\mathcal{O}^{\mathsf{KG}}$, $\mathcal{O}^{\mathsf{Aut}_1}$, $\mathcal{O}^{\mathsf{Aut}_2}$, $\mathcal{O}^{\mathsf{Dec}}$ are performed as follows:

- $\mathcal{O}^{\mathsf{KG}}$: Given an index $i \neq t$ of user U_i, it responds (pk_i, sk_i) of user U_i, which is the outcome of $\mathsf{KeyGen}(pp)$.
- $\mathcal{O}^{\mathsf{Aut}_1}$: Given an index i, it responds a token for user U_i that is the outcome of $\mathsf{Aut}(sk_i)$.

- $\mathcal{O}^{\mathsf{Aut}_2}$: Given a pair of index and ciphertext (i, CT_i), it responds a token that is the outcome of $\mathsf{Aut}(sk_i, CT_i)$.
- $\mathcal{O}^{\mathsf{Dec}}$: Given a pair of index and ciphertext (i, CT_i), it responds a plaintext M_i that is the same as the outcome of $\mathsf{Dec}(sk_i, CT_i)$.

For $\mathcal{A}$'s queries, there are several restrictions as follows:

- The index of the target user does not appear in $\mathcal{O}^{\mathsf{KG}}$.
- The pair of the target user's index and the target ciphertext does not appear in $\mathcal{O}^{\mathsf{Dec}}$.

The advantage of $\mathcal{A}$ in $\mathbf{Exp}^{\text{OW-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$ is defined as

$$\mathbf{Adv}^{\text{OW-CCA2}}_{\mathcal{A},\text{PKE-AET}}(\lambda) = \Pr[M = M'].$$

In PKEET schemes, if the plaintext space is not sufficiently large or the plaintext distribution has special shapes, then the adversary who gets a token for equality test may attempt at performing trivial attacks by generating ciphertexts and then performing equality tests. To avoid such type of attacks, we give a remark the conditions on the plaintext space and distribution.

Remark 1: As other literature in the PKEET area, the cardinality of the plaintext space should be exponential in λ. In addition, we also assume that the min-entropy of plaintext distribution is sufficiently higher than λ. If the above conditions do not hold, the adversary who has a token can break the OW-CCA2 security by generating ciphertexts of messages and then conducting equality tests between generated ciphertext and the target ciphertext.

Next, the IND-CCA2 security against a Type-II adversary is formally defined below.

Definition 4 (IND-CCA2 Security against Type-II Adversaries): A PKE-AET scheme is IND-CCA2 secure against Type-II adversaries if there is no PPT adversary $\mathcal{A}$ whose advantage in the experiment $\mathbf{Exp}^{\text{IND-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$, defined as below, is negligible in the security parameter λ:

```
$\mathbf{Exp}^{\text{IND-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$
$\mathsf{Setup}(\lambda) \to (pp)$
$\mathsf{KeyGen}(pp) \to (pk_t, sk_t)$
$\mathcal{A}^{\mathcal{O}^{\mathsf{KG}},\mathcal{O}^{\mathsf{Aut}_1},\mathcal{O}^{\mathsf{Aut}_2},\mathcal{O}^{\mathsf{Dec}}}(pk_t) \to M_0, M_1$
$\beta \xleftarrow{\$} \{0,1\}$
$\mathsf{Enc}(pk_t, M_\beta) \to CT^*_\beta$
$\mathcal{A}^{\mathcal{O}^{\mathsf{KG}},\mathcal{O}^{\mathsf{Aut}_1},\mathcal{O}^{\mathsf{Aut}_2},\mathcal{O}^{\mathsf{Dec}}}(CT^*_\beta) \to \beta'$
```

The oracles $\mathcal{O}^{\mathsf{KG}}, \mathcal{O}^{\mathsf{Aut}_1}, \mathcal{O}^{\mathsf{Aut}_2}, \mathcal{O}^{\mathsf{Dec}}$ utilized in the above experiment are defined as the same as those of the experiment $\mathbf{Exp}^{\text{OW-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$ in Definition 3. In this experiment, there are same constraints on $\mathcal{A}$'s queries as in the previous game. In addition, the target user's index does not appear in $\mathcal{O}^{\mathsf{Aut}_1}$.

The advantage of $\mathcal{A}$ in $\mathbf{Exp}^{\text{IND-CCA2}}_{\text{PKE-AET},\mathcal{A}}(\lambda)$ is defined to

$$\mathbf{Adv}^{\text{IND-CCA2}}_{\mathcal{A},\text{PKE-AET}}(\lambda) = \left| \Pr[\beta = \beta'] - \frac{1}{2} \right|.$$

B. Underlying Assumptions

We will prove the correctness and security of our PKE-AET scheme under the hardness of the RSA problem and some features of hash functions. We first introduce the formal definitions of the RSA problem and assumption below.

Definition 5 (RSA Assumption): Let N be an RSA modulus where $N = pq$ with two primes p, q. Given N, an exponent e in $\mathbb{Z}^*_{\phi(N)}$, and a target value $C \in \mathbb{Z}_N$, the RSA problem is to compute M such that

$$C = M^e \bmod N.$$

It is said that the RSA assumption holds for (N, e) if there is no PPT algorithm $\mathcal{A}$ whose advantage defined to

$$\mathbf{Adv}^{\mathsf{RSA}}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A}(N, e, C) \to M : \\ M^e \equiv C \bmod N, C \in_R \mathbb{Z}^*_N]$$

is negligible in λ where N is a randomly generated RSA modulus and e is an exponent randomly chosen from $\mathbb{Z}^*_{\phi(N)}$.

Next, we provide two definitions for features of hash functions.

Definition 6 (One-Wayness of Hash Functions): It is said that a hash function $H : \{0,1\}^* \to \{0,1\}^\ell$ with $\ell = \ell(\lambda)$ for $\lambda \in \mathbb{Z}$ is one-way if there is no PPT algorithm $\mathcal{A}$ whose advantage defined to

$$\Pr[\mathcal{A}(H, y) \to m : H(m) = y \text{ and } y \in_R \{0,1\}^\ell]$$

is negligible in λ.

Definition 7 (Collision Resistance of Hash Functions): It is said that a hash function $H : \{0,1\}^* \to \{0,1\}^\ell$ with $\ell = \ell(\lambda)$ for $\lambda \in \mathbb{Z}$ is collision-resistant if there is no PPT algorithm $\mathcal{A}$ whose advantage defined to

$$\Pr[\mathcal{A} \to (m, m') : H(m) = H(m') \text{ and } m \neq m']$$

is negligible in λ.

III. Our New RSA-Based PKE-AET

Now, we proivde our new PKE-AET construction based on the RSA assumption. Then, we analyze the proposed construction in terms of correctness and security.

A. Our New PKE-AET Construction from the RSA Assumption

We first present the description of our RSA-based PKE-AET construction.

Description of Our PKE-AET. Our RSA-based PKE-AET scheme consists of 7 polynomial time algorithms. Below, we give the full description of our new PKE-AET scheme.

- $\mathsf{Setup}(\lambda)$: It takes a security parameter λ as an input, select three hash functions:
  - $H_1 : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^{2\lambda}$,
  - $H_2 : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda}$, and
  - $H_3 : \{0,1\}^\ell \times \{0,1\}^\ell \times \{0,1\}^\ell \times \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda}$

where $\ell = \ell(\lambda)$ is the bit length of N_1 and N_2 each, which will be generated by the KeyGen algorithm. Then, it returns the public parameter $pp = (H_1, H_2, H_3)$.

- $\mathsf{KeyGen}(pp)$: Given $pp = (H_1, H_2, H_3)$, it does as the followings:
    1) Pick four $(\ell/2)$-bit prime numbers p_1, q_1, p_2, q_2.
    2) Compute $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$.
    3) Compute $\phi(N_1) = (p_1 - 1)(q_1 - 1)$ and $\phi(N_2) = (p_2 - 1)(q_2 - 1)$ for the Euler totient function ϕ.
    4) Select two elements e_1 and e_2 from $\mathbb{Z}^*_{\phi(N_1)}$ and $\mathbb{Z}^*_{\phi(N_1)}$, respectively.
    5) Compute d_1 and d_2 such that
    $$e_1 d_1 \equiv 1 \bmod \phi(N_1) \quad \text{and} \quad e_2 d_2 \equiv 1 \bmod \phi(N_2).$$
    6) Return the public key $pk = (N_1, N_2, e_1, e_2)$ and the secret key $sk = (N_1, N_2, d_1, d_2)$.
- $\mathsf{Enc}(pk, M)$: On input $pk = (N_1, N_2, e_1, e_2)$ and a plaintext $M \in \{0,1\}^{2\lambda}$, it performs as follows:
    1) Choose r_1 and r_2 randomly from $\mathbb{Z}_{N_1}$ and $\mathbb{Z}_{N_2}$, respectively.
    2) Calculate
        a) $C_1 = r_1^{e_1} \bmod N_1$,
        b) $C_2 = r_2^{e_2} \bmod N_2$,
        c) $C_3 = M \oplus H_1(r_1, r_2)$, and
        d) $C_4 = H_2(M) \oplus H_3(r_2, C_1, C_2, C_3)$
    3) Output a ciphertext $CT = (C_1, C_2, C_3, C_4)$.
- $\mathsf{Dec}(sk, CT)$: Given $sk = (N_1, N_2, d_1, d_2)$ and a ciphertext $CT = (C_1, C_2, C_3, C_4)$, it performs the followings:
    1) Calculate
        a) $r'_1 = C_1^{d_1} \bmod N_1$,
        b) $r'_2 = C_2^{d_2} \bmod N_2$,
        c) $M' = C_3 \oplus H_1(r'_1, r'_2)$, and
        d) $h' = C_4 \oplus H_3(r'_2, C_1, C_2, C_3)$.
    2) Check if $h' = H_2(M')$. If it holds, return M'. Return $\perp$, otherwise.
- $\mathsf{Aut}_1(sk_i)$: On input $sk_i = (N_{i,1}, N_{i,2}, d_{i,1}, d_{i,2})$ of user U_i, it returns $tk_{i,1} = (N_{i,2}, d_{i,2})$.
- $\mathsf{Aut}_2(sk_i, CT_i)$: Given $sk_i = (N_{i,1}, N_{i,2}, d_{i,1}, d_{i,2})$ and $CT_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4})$ of user U_i,
    1) Compute $r_{i,2} = C_{i,2}^{d_{i,2}} \bmod N_{i,2}$.
    2) Compute and output
    $$tk_{i,2,CT_i} = H_3(r_{i,2}, C_{i,1}, C_{i,2}, C_{i,3}).$$
- $\mathsf{Test}(CT_A, tk_A, CT_B, tk_B)$: Given two pairs of ciphertext and token of users U_A and U_B, respectively, it performs as follows: Let $CT_A = (C_{A,1}, C_{A,2}, C_{A,3}, C_{A,4})$ and $CT_B = (C_{B,1}, C_{B,2}, C_{B,3}, C_{B,4})$.
    1) For (CT_A, tk_A),
        - If $tk_A = (N_{A,2}, d_{A,2})$ is the output of Aut_1 algorithm,
            a) Compute $r_{A,2} = C_{A,2}^{d_{A,2}} \bmod N_{A,2}$,
            b) Compute $h_A = C_{A,4} \oplus H_3(r_{A,2}, C_{A,1}, C_{A,2}, C_{A,3})$.
        - If tk_A is the output of Aut_2 algorithm, compute $h_A = tk_A \oplus C_{A,4}$.
    2) For (CT_B, tk_B), compute h_B as in Step 1).
    3) Check if $h_A = h_B$. Output 1, if it holds and 0, otherwise.

Correctness of Our PKE-AET. We investigate the correctness of the proposed PKE-AET construction in Theorem 1.

Theorem 1: If the exploited hash function H_3 is collision resistant, then the PKE-AET construction presented in this section is correct.

Proof: Suppose that $CT = (C_1, C_2, C_3, C_4)$ is a ciphertext of plaintext M which is generated by the Enc algorithm with inputs pk and M where pk is obtained by executing $\mathsf{Setup}(\lambda) \to pp$ and $\mathsf{KeyGen}(pp) \to (pk, sk)$ sequentially. Then, CT is

$$\begin{aligned} C_1 &= r_1^{e_1} \bmod N_1, \\ C_2 &= r_2^{e_2} \bmod N_2, \\ C_3 &= M \oplus H_1(r_1, r_2), \text{ and} \\ C_4 &= H_2(M) \oplus H_3(r_2, C_1, C_2, C_3) \end{aligned}$$

where r_1, r_2 are randomly selected by the Enc algorithm and $pk = (N_1, N_2, e_1, e_2)$.

In the Dec algorithm, if CT is given with the secret key $sk = (N_1, N_2, d_1, d_2)$ which is corresponded to $pk = (N_1, N_2, e_1, e_2)$,

$$\begin{aligned} r'_1 &= C_1^{d_1} = (r_1^{e_1})^{d_1} = r_1^{e_1 d_1} = r_1 \bmod N_1 \text{ and} \\ r'_2 &= C_2^{d_2} = (r_2^{e_2})^{d_2} = r_2^{e_2 d_2} = r_2 \bmod N_2 \end{aligned} \tag{1}$$

hold from the Euler theorem. So, the Dec algorithm recovers r_1, r_2 correctly. Thus,

$$\begin{aligned} M' &= C_3 \oplus H_1(r'_1, r'_2) \\ &= M \oplus H_1(r_1, r_2) \oplus H_1(r'_1, r'_2) = M \text{ and} \\ h' &= C_4 \oplus H_3(r'_2, C_1, C_2, C_3) \\ &= H_2(M) \oplus H_3(r_2, C_1, C_2, C_3) \oplus H_3(r'_2, C_1, C_2, C_3) \\ &= H_2(M) \end{aligned}$$

hold since $r_1 = r'_1$ and $r_2 = r'_2$. Therefore, $h' = H_2(M) = H_2(M')$ holds and the Dec algorithm returns M correctly.

Next, suppose that $CT_A = (C_{A,1}, C_{A,2}, C_{A,3}, C_{A,4})$ and $CT_B = (C_{B,1}, C_{B,2}, C_{B,3}, C_{B,4})$ are two ciphertexts of messages M_A and M_B generated by running $\mathsf{Enc}(pk_A, M_A)$ and $\mathsf{Enc}(pk_B, M_B)$ where pk_A and pk_B are public keys of users U_A and U_B, respectively. Once tokens tk_A and tk_B are given, if tk_A or tk_B is a token for all ciphertexts of user U_A or U_B, respectively, that is, $tk_A = (N_{A,2}, d_{A,2})$ and $tk_B = (N_{B,2}, d_{B,2})$, then the tester obtains $r_{A,2}$ or $r_{B,2}$ correctly by computing

$$r_{A,2} = C_{A,2}^{d_{A,2}} \bmod N_{A,2} \quad \text{or} \quad r_{B,2} = C_{B,2}^{d_{B,2}} \bmod N_{B,2},$$

TABLE I
FEATURE COMPARISON OF RSA-BASED PKEET AND PKE-AET

		[9]	Instantiation from [11]	Ours
Test Type	Aut_1	✓	✓	✓
	Aut_2	✗	✗	✓
Security	Type-I	OW-CCA1	OW-CCA2	OW-CCA2
	Type-II	IND-CCA1	IND-CCA2	IND-CCA2

respectively, from Eq. (1). Then, the tester obtains h_A or h_B by computing

$$\begin{aligned} h_A &= C_{A,4} \oplus H_3(r_{A,2}, C_{A,1}, C_{A,2}, C_{A,3}) \text{ or} \\ h_B &= C_{B,4} \oplus H_3(r_{B,2}, C_{B,1}, C_{B,2}, C_{B,3}), \end{aligned}$$

which corresponds to $H_2(M_A)$ or $H_2(M_B)$, respectively.

Otherwise, if tk_A or tk_B is a token for a particular ciphertext CT_A or CT_B of user U_A or U_B, respectively, that is, $tk_A = H_3(r_{A,2}, C_{A,1}, C_{A,2}, C_{A,3})$ and $tk_B = H_3(r_{B,2}, C_{B,1}, C_{B,2}, C_{B,3})$, then the tester obtains h_A or h_B correctly by computing

$$\begin{aligned} h_A &= C_{A,4} \oplus tk_A = C_{A,4} \oplus H_3(r_{A,2}, C_{A,1}, C_{A,2}, C_{A,3}) \text{ or} \\ h_B &= C_{B,4} \oplus tk_B = C_{B,4} \oplus H_3(r_{B,2}, C_{B,1}, C_{B,2}, C_{B,3}), \end{aligned}$$

respectively.

Thus, for any type of tokens, the tester obtains $h_A = H_2(M_A)$ and $h_B = H_2(M_B)$. Therefore, the tester always returns 1 if $M_A = M_B$ and returns 0 without negligible probability if $M_A \neq M_B$ and H_2 is collision-resistant. ■

B. Security of Our PKE-AET Scheme

Now, we take a look at the security of our construction, described in III-A. The below two theorems investigate the OW-CCA2 and IND-CCA2 security of the proposed construction against Type-I PPT adversaries and Type-II PPT adversaries, respectively. We omit the full proofs of those theorems and give proof sketches of them, due to the space constraints.

Theorem 2 (OW-CCA2 Security): Under the RSA assumption and the onewayness of the employed hash function H_2, our PKE-AET scheme, given in Section III-A, is OW-CCA2 secure against any Type-I PPT adversary in the random oracle model.

Proof of Sketch. In the security proof of Theorem 2, suppose that there is a PPT adversary $\mathcal{A}_1$ breaking the OW-CCA2 security of our PKE-AET scheme. Then, we design an algorithm $\mathcal{B}_1$ that resolves the RSA problem with an instance (N, e, C) by interacting with $\mathcal{A}_1$ as the challenger $\mathcal{C}$ in the OW-CCA2 security game.

In the setup phase of the security game, $\mathcal{B}_1$ embeds the RSA instance into the public key of the target user U_t, that is, $\mathcal{B}_1$ sets $N_{t,1} = N$ and $e_{t,1} = e$, and passes the public key of target user U_t, $pk_t = (N_{t,1}, N_{t,2}, e_{t,1}, e_{t,2})$, to $\mathcal{A}_1$. Then, at the challenge phase, $\mathcal{B}_1$ generates the challenge ciphertext as

$$\begin{aligned} &C_1^* = C, \\ &C_2^* = (r_2^*)^{e_{t,2}} \bmod N_{t,2}, \\ &C_3^* = R_3^*, \text{ and} \\ &C_4^* = H_2(M) \oplus H_3(r_2^*, C_1^*, C_2^*, C_3^*) \end{aligned}$$

where r_2^* and R_3^* are randomly selected from $\mathbb{Z}_{N_{t,2}}$ and $\{0,1\}^{2\lambda}$, respectively, after selecting a random target message M. Then, after finishing the guess phase, once $\mathcal{A}_1$ outputs M', if $M' = M$, then $\mathcal{B}_1$ can find r such that $C = r^e \bmod N$ by searching the tables for hash queries. Thus, $\mathcal{B}_1$ can resolve the RSA problem with at least the advantage of $\mathcal{A}_1$. We note that $\mathcal{B}_1$ can respond to $\mathcal{A}_1$'s queries appropriately by using the hash tables for hash queries since random oracle heuristics are assumed.

Theorem 3 (IND-CCA2 Security): Under the RSA assumption, our PKE-AET construction, presented in Section III-A, is IND-CCA2 secure against any Type-II PPT adversary in the random oracle model.

Similarly to Theorem 2, we can prove Theorem 3 by constructing an algorithm $\mathcal{B}_2$ that resolves the RSA problem with an instance (N, e, C) by interacting with $\mathcal{A}_2$ as the challenger $\mathcal{C}$ in the IND-CCA2 security game where $\mathcal{A}_2$ is an adversary breaking the IND-CCA2 security of our PKE-AET scheme. We omit the proof of Theorem 3 as well, due to the space constraints.

IV. COMPARISON OF OUR AND OTHER EXISTING RSA-BASED CONSTRUCTIONS

Now, we compare our construction with other existing RSA-based PKEET schemes in terms of features and performance.

A. Feature Comparison

We first present compare features of our proposed construction and other RSA-based PKEET schemes. In Table I, the third, fourth, and fifth columns provide main characteristics of Zhu et al.'s scheme [9], a concrete RSA-based instantiation from the generic construction in [11] with RSA-OAEP [12], and our proposed construction, respectively. According to the table, the instantiation from the generic construction and our proposed construction are secure against CCA2, while Zhu et al.'s construction achieves CCA1 security only. Furthermore, only our proposed construction supports issuing a token for a particular ciphertext as well as for all ciphertexts of user,

TABLE II
PERFORMANCE COMPARISON OF RSA-BASED PKEET AND PKE-AET

	λ	KeyGen	Enc	Dec	Aut_1	Aut_2	Test	
		(s)	(ms)	(ms)	(μs)	(ms)	w/ $tk_{i,1}$	w/ $tk_{i,2,CT}$
							(ms)	(μs)
[9]	80	0.332	0.580	0.801	0.272	–	0.546	–
	112	4.636	3.431	5.070	0.211	–	3.475	–
	128	23.779	10.857	16.175	0.290	–	10.691	–
Instantiation from [11]	80	0.335	0.539	0.531	0.374	–	0.521	–
	112	4.363	3.489	3.482	0.243	–	3.428	–
	128	23.195	10.791	10.865	0.272	–	10.532	–
Ours	80	0.350	0.532	0.510	0.413	0.255	0.512	3.238
	112	4.653	3.321	3.327	0.494	1.703	3.554	3.570
	128	24.139	10.791	10.576	0.490	5.455	10.751	3.897

while other two constructions support issuing a token for all ciphertexts of user only.

B. Performance Comparison

Now, we present implementation results of our construction and other two RSA-based PKEET schemes. Our source codes were written in C++ and used OpenSSL library [13] for large number operations and hash function computations. The test was done on a PC with an Intel(R) Core(TM) i7-11700 CPU running at 2.50 GHz and 32 GB RAM.

Table II shows experimental results of our proposed construction and other RSA-based PKEET construction with respect to security levels. We generated RSA moduli of 1024, 2048, and 3072 bits for 80, 112, and 128 bits security levels, respectively, by following the NIST recommendation. Our experimental results demonstrate that all algorithms of our proposed scheme have comparable efficiency to those of other existing RSA-based PKEET constructions, while supporting additional types of equality tests. Particularly, ours and the concrete RSA-based instantiation from the generic construction in [11] outperforms Zhu et al.'s construction with regard to the execution time of the decryption algorithm with achieving the enhanced security.

V. CONCLUSION

This article proposed a new RSA-based PKE-AET scheme, which enables the owner of ciphertexts to issue a token for a particular ciphertext as well as all ciphertexts of user. The proposed construction is OW-CCA2 secure against any Type-I PPT adversary who may get tokens and IND-CCA2 secure against any Type-II PPT adversary who cannot get under the RSA assumption in the random oracle model. Subsequently, we demonstrate that our proposed scheme has comparable efficiency to other RSA-based PKEET schemes by presenting experimental results of ours and others under various parameters.

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 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, ser. Lecture Notes in Computer Science, J. Pieprzyk, Ed., vol. 5985. Springer, 2010, pp. 119–131.

[2] Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," *Secur. Commun. Networks*, vol. 5, no. 12, pp. 1351–1362, 2012.

[3] ——, "Public key encryption schemes supporting equality test with authorisation of different granularity," *Int. J. Appl. Cryptogr.*, vol. 2, no. 4, pp. 304–321, 2012.

[4] 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, 2015.

[5] S. Ma, Q. Huang, M. Zhang, and B. Yang, "Efficient public key encryption with equality test supporting flexible authorization," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 3, pp. 458–470, 2015.

[6] K. Huang, R. Tso, Y. 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, 2015.

[7] W. Susilo, D. H. Duong, and H. Q. Le, "Efficient post-quantum identity-based encryption with equality test," in *26th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2020*. IEEE, 2020, pp. 633–640.

[8] 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, 2022.

[9] H. Zhu, D. Xie, H. Ahmad, and H. N. Hasan Abdullah, "New constructions of equality test scheme for cloud-assisted wireless sensor networks," *PLOS ONE*, vol. 16, no. 10, pp. 1–15, 10 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0258746

[10] X. J. Lin, L. Sun, and H. Qu, "Generic construction of public key encryption, identity-based encryption and signcryption with equality test," *Inf. Sci.*, vol. 453, pp. 111–126, 2018.

[11] 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, 2019.

[12] V. Shoup, "OAEP reconsidered," *J. Cryptol.*, vol. 15, no. 4, pp. 223–249, 2002.

[13] "OpenSSL–Cryptography and SSL/TLS Toolkit, Version 1.1.1n," Available at https://www.openssl.org, 2022, online; Accessed 7 Apr 2022.