Public Key Authenticated Encryption with Designated Equality Test and its Applications in Diagnostic Related Groups

Yuanhao Wang  
*South China Agricultural University*

Qiong Huang  
*South China Agricultural University*, csqhuang@alumni.cityu.edu.hk

Hongbo Li  
*South China Agricultural University*

Jianye Huang  
*University of Wollongong*, jh207@uowmail.edu.au

Guomin Yang  
*University of Wollongong*, gyang@uow.edu.au

See next page for additional authors

Follow this and additional works at: [https://ro.uow.edu.au/eispapers1](https://ro.uow.edu.au/eispapers1)

Part of the Engineering Commons, and the Science and Technology Studies Commons

Recommended Citation

Wang, Yuanhao; Huang, Qiong; Li, Hongbo; Huang, Jianye; Yang, Guomin; and Susilo, Willy, "Public Key Authenticated Encryption with Designated Equality Test and its Applications in Diagnostic Related Groups" (2019). *Faculty of Engineering and Information Sciences - Papers: Part B*. 3254.  
[https://ro.uow.edu.au/eispapers1/3254](https://ro.uow.edu.au/eispapers1/3254)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Public Key Authenticated Encryption with Designated Equality Test and its Applications in Diagnostic Related Groups

Abstract
Due to the massive growth of data and security concerns, data of patients would be encrypted and outsourced to the cloud server for feature matching in various medical scenarios, such as personal health record systems, actuarial judgements and diagnostic related groups. Public key encryption with equality test (PKEET) is a useful utility for encrypted feature matching. Authorized tester could perform data matching on encrypted data without decrypting. Unfortunately, due to the limited terminology in medicine, people within institutions may illegally use data, trying to obtain information through traversal methods. In this paper we propose a new PKEET notion, called public-key authenticated encryption with designated equality test (PKAE-DET), which could resist this kind of attacks launched by an inside adversary, known as offline message recovery attacks (OMRA). We propose a concrete construction of PKAE-DET, which only requires one single server to perform the feature matching job securely, and does not require any group mechanism. We prove its security based on some simple mathematical assumptions. Experimental results show that our scheme has efficiency comparable with those PKEET schemes which do not resist OMRA attacks or require group mechanism. We further show how our scheme could be effectively used in diagnostic related groups in medicine, demonstrating its practicability.

Keywords
key, public, test, equality, its, designated, applications, diagnostic, related, groups, encryption, authenticated

Disciplines
Engineering | Science and Technology Studies

Publication Details
Wang, Y., Huang, Q., Li, H., Huang, J., Yang, G. & Susilo, W. (2019). Public Key Authenticated Encryption with Designated Equality Test and its Applications in Diagnostic Related Groups. IEEE Access, 7 135999-136011.

Authors
Yuanhao Wang, Qiong Huang, Hongbo Li, Jianye Huang, Guomin Yang, and Willy Susilo

This journal article is available at Research Online: https://ro.uow.edu.au/eispapers1/3254
Public Key Authenticated Encryption With Designated Equality Test and Its Applications in Diagnostic Related Groups

YUANHAO WANG1, QIONG HUANG2, (Member, IEEE), HONGBO LI1, JIANYE HUANG2, GUOMIN YANG2, (Senior Member, IEEE), AND WILLY SUSILO2, (Senior Member, IEEE)

1College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
2School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia

Corresponding author: Qiong Huang (qhuang@scau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872152 and Grant 61872409, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2014A030306021, in part by the Guangdong Program for Special Support of Top-notch Young Professionals under Grant 2015TQ01X796, and in part by the Graduate Student Overseas Study Program of South China Agricultural University under Grant 2018LHPY025.

ABSTRACT Due to the massive growth of data and security concerns, data of patients would be encrypted and outsourced to the cloud server for feature matching in various medical scenarios, such as personal health record systems, actuarial judgments and diagnostic related groups. Public key encryption with equality test (PKEET) is a useful utility for encrypted feature matching. Authorized tester could perform data matching on encrypted data without decrypting. Unfortunately, due to the limited terminology in medicine, people within institutions may illegally use data, trying to obtain information through traversal methods. In this paper we propose a new PKEET notion, called public-key authenticated encryption with designated equality test (PKAE-DET), which could resist this kind of attacks launched by an inside adversary, known as offline message recovery attacks (OMRA). We propose a concrete construction of PKAE-DET, which only requires one single server to perform the feature matching job securely, and does not require any group mechanism. We prove its security based on some simple mathematical assumptions. Experimental results show that our scheme has efficiency comparable with those PKEET schemes which do not resist OMRA attacks or require group mechanism. We further show how our scheme could be effectively used in diagnostic related groups in medicine, demonstrating its practicability.

INDEX TERMS Public key encryption, equality test, message recovery attack, cloud storage, data sharing.

I. INTRODUCTION

The development of cloud computing technology provides more efficient and reliable data management with less error to us, and the service quality of traditional industries has been effectively improved. One of the most prominent manifestations is cloud-based electronic medical systems [1], [2].

Diagnostic Related Groups (DRG) constitutes an approach to measuring hospital case mix, which may be understood as a system for separating hospitalized patients into unique groups based on their diagnoses and procedures [3]. According to diagnoses, procedures, age, sex, discharge status, and the presence of complications or comorbidities, patients are divided into different diagnosis related groups, which are scientifically calculated on the basis of classification and given an advanced payment. This kind of payment method takes care of the interests of patients, hospitals, medical insurances and other aspects, and becomes the most important cost control and quality improvement tool that governments and private payers have implemented [4].

Due to the rapid growth of medical data and low maintenance cost demand, hospitals would prefer to store patients’ medical records onto cloud servers. However, cloud servers are usually provided by third-party providers rather than hospitals themselves. Due to the high sensitiveness and privacy of patients’ data, hospitals would encrypt them before...
uploading. However, it brings new challenges. Once the data is encrypted, its structure is completely hidden. Search operations over plaintext becomes invalid.

Searchable encryption (SE) was thus proposed to solve this problem effectively, and has attracted wide attentions since its introduction. It allows users to store a large group of encrypted documents on the cloud server and to perform keyword-based searches over these encrypted documents. The server cannot get further information except for specific patterns. SE could be divided into two categories: symmetric searchable encryption (SSE) [5] and public-key encryption with keyword search (PEKS) [6].

In 2010 a new encryption primitive named public-key encryption with equality test (PKEET) [7] was introduced, which could be viewed as a variant of PEKS. PKEET is designed to be used in specific situations, since it allows the tester to check whether two ciphertexts generated under different public keys contain the same plaintext without decryption.

As shown in Figure 1, PKEET can be used in DRG systems, personal health record systems, and other medical systems. (Please refer to Sect. VII for a more detailed explanation.) Since patients within each category are clinically similar and are expected to use the same level of hospital resources [8], patients’ medical records can be encrypted and compared with the encrypted criterions in the system to determine which category the patient’s illness should correspond to. However, due to privacy issue, it is necessary to prevent outside or inside personnel from illegally using the data for operations beyond the authorization. Compared with outside adversaries, it is harder to guard patients’ data against inside adversaries, which are able to launch offline message recovery attacks (OMRA).

A. OFFLINE MESSAGE RECOVERY ATTACK

Offline message recovery attack (OMRA) is an inside attack proposed by Tang [9], in which an inside adversary would be able to recover the plaintext from a given ciphertext. The inside adversary refers an insider who can easily obtain the trapdoor, such as a database administrator in the hospital. It wants to obtain the patient’s privacy information in order to gain benefits.

Figure 2 shows the flow chart of OMRA. To recover the message $m$ encapsulated in ciphertext $C$ w.r.t. public key $pk$, an OMRA adversary, which is given a trapdoor $td$ as the authorization to test over $C$, works as follows:

1) generates a new key pair $(pk', sk')$;
2) randomly chooses a new message $m'$ from the message space, and encrypted $m'$ under $pk'$ to get ciphertext $C'$;
3) uses $sk'$ to produce a trapdoor $td'$ w.r.t. $pk'$;
4) runs the test algorithm on input $(td, C, td', C')$ to check if $C$ and $C'$ contain the message; and
5) outputs $m'$ if the test returns 1, and goes to Step 2) otherwise.

Theoretically, we assume the plaintext space to be at least super-logarithmically large. However, in practice, especially in medicine, all data are certain medical words, and the frequency of each word is uneven. The occurrence probabilities of some common disease-related words are high, and the others are low. Therefore, it is feasible for an inside adversary to launch and finish the attack above in a reasonable time and recovers $m$ from $C$, even if the adversary has limited computational power.

In this regard, Tang [9] proposed a PKEET scheme that can resist OMRA attack. However, their scheme needs to work in dual-server setting, which requires high computation and communication costs, and assumes that the two servers do
not collude. Therefore, it is necessary to construct a PKEET scheme that is secure against OMRA attack and works in the single-server setting.

B. RELATED WORKS

1) PKEET
The notion of PKEET was introduced by Yang et al. [7] in CT-RSA 2010, which extended public key encryption by adding a test algorithm to compare the data encrypted under different public keys without decryption. It has attracted extensive attentions. Since the scheme of Yang et al. [7] has no authorization mechanism, everyone could compare encrypted data. Tang et al. [10] proposed an FG-PKEET scheme, in which the server can only perform the ciphertext comparison after obtaining the trapdoor generated by two users together. Subsequently, Tang et al. proposed an AoN-PKEET scheme [11] and an ADG-PKEET scheme [9] to improve the authorization mechanism. The former requires that the test obtains trapdoors from both users, with which it can compare all ciphertexts of them. The latter is an extension of FG-PKEET, which resists OMRA attacks via the dual-server method, but needs to make double agent agreement to interact, and thus requiring higher computation costs. Ma et al. [12] proposed a PKE-DET scheme by delegating the comparison to a third party to ensure that only the designated server could use the trapdoor to compare ciphertexts. Huang et al. [13] proposed a PKE-AET scheme, which supports two authorization patterns, e.g. comparing either all ciphertexts or some specific ciphertexts of users, respectively. Ma et al. [14] introduced PKEET-FA scheme, supporting four types of authorization patterns, e.g. comparing either all ciphertexts or some specific ciphertexts of users, respectively. Ma et al. [15] proposed PKAE-DET scheme, which supports two other authorization patterns different from those in [14], and in the meanwhile supports the verification of calculated results.

In order to solve the problem of certificate management, [16]–[20] proposed schemes of identity-based encryption with equality test (IBEET) by combining PKEET and IBE. Besides, [21]–[24] proposed schemes of attribute-based encryption with equality test (ABEET) in order to achieve more flexible authorization, which is a combination of PKEET and ABE. These researches increase the security and further extend the application scenarios of PKEET.

2) OMRA AND IKGA
Tang [9] proposed to use dual servers to resist OMRA attacks. The same method is also used in [25], [26]. It assumes that the servers do not collude and requires additional communication between the servers. Relatively, [27], [28] proposed to share a secret among group members only with which can a member generate a valid ciphertext legally. It is more efficient than the aforementioned dual-server schemes. However, the group creation takes time, and there is a risk of leaking the shared secret. If a group member colludes with the server, they can still launch the OMRA attack.

OMRA is similar with the inside keyword guessing attack (IKGA) against public key searchable encryption. In both of the attacks, a malicious server with the trapdoor tries to guess the message from a given ciphertext. Schemes secure against IKGA can be roughly divided into two categories. The first category [29]–[31] works in the dual-server setting, but requires additional communication overhead. The second category [32], [33] adopts to use the sender’s private key to encrypt a message. In all these schemes (of the second category), the sender is also able to generate a trapdoor. It would cause security issues if we directly apply this method in PKEET.

C. OUR CONTRIBUTIONS
In this work we study how to construct an efficient PKEET scheme resisting OMRA attacks in the single-server setting. Concretely, we make the following contributions:

1) We propose a new notion of public-key authenticated encryption with designated equality test (PKAE-DET), which to the best of our knowledge, is the first PKEET scheme that is secure against OMRA attacks and does not require the dual-server setting or the group mechanism. In PKAE-DET, the sender specifies who is responsible to test ciphertexts while encrypting a message, in order to prevent outside adversaries from testing. The receiver specifies the sender and the target user to compare with while generating the trapdoor, in order to prevent inside adversaries from abusing the authorization.

2) We propose a concrete construction of PKAE-DET which makes use of an asymmetric bilinear pairing, and prove its security based on simple mathematical assumptions (e.g. (decisional) Bilinear Diffie-Hellman assumption, symmetric external Diffie-Hellman assumption) under the given security models. Experimental results show that our scheme has efficient comparable with previous PKEET schemes.

3) We further show how our PKAE-DET scheme could be used in applications like DRG, thus demonstrating the usability and practicability of our scheme.

D. PAPER ORGANIZATION
The rest of this paper is organized as follows. In Sect. II, we introduce some cryptographic preliminaries. In Sect. III we describe the formal definition and security models of PKAE-DET. The concrete construction is proposed in Sect. IV, and its security analysis is provided in Sect. V. In Sect. VI we present the performance evaluation and comparison of our PKAE-DET scheme and other related PKEET schemes. We then show the application of our scheme in DRG in Sect. VII, and conclude the paper in Sect. VIII.

II. PRELIMINARIES

A. BILINEAR PAIRING
Let \( G_1, G_2 \) and \( G_T \) be groups of prime order \( p \). A bilinear pairing \( \hat{e} : G_1 \times G_2 \rightarrow G_T \) satisfies the following properties.
Bilinearity: For any \( X \in \mathbb{G}_1, Y \in \mathbb{G}_2 \) and any \( a, b \in \mathbb{Z}, \hat{e}(X^a, Y^b) = \hat{e}(X, Y)^{ab} \).

Non-degeneracy: If \( g \) and \( h \) are generators of \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \), respectively, \( \hat{e}(g, h) \neq 1_{\mathbb{G}_1} \).

Computability: There is an efficient algorithm to compute \( \hat{e}(X, Y) \) for any \( X \in \mathbb{G}_1, Y \in \mathbb{G}_2 \).

B. BILINEAR DIFFIE-HELLMAN (BDH) ASSUMPTION

Let \( \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \), \( \hat{e} \) and \( p \) be defined as above, and \( g_1, g_2 \) be generators of \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \), respectively. Given the tuple \((g_1, g_2, g_1^a, g_2^b, g_1^c)\) where \( a, b, c \overset{\$}{\leftarrow} \mathbb{Z}_p^* \), we say BDH assumption holds if for any probabilistic polynomial-time (PPT) algorithm \( A \), it computes \( \hat{e}(g_1, g_2)^{abc} \) with only negligibly probability.

C. SYMMETRIC external DIFFIE-HELLMAN (SXDH) ASSUMPTION

Given a tuple \((g, g^a, g^b, g^c)\) where \( a, b, r \overset{\$}{\leftarrow} \mathbb{Z}_p^* \), we say decisional Diffie-Hellman (DDH) assumption holds if for any PPT algorithm \( A \), it decides whether \( g^r = g^{ab} \) with probability only negligibly larger 1/2. The SXDH assumptions holds if DDH assumption holds in both \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \).

D. DECISIONAL BILINEAR DIFFIE-HELLMAN (DBDH) ASSUMPTION

Let \( \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \), \( \hat{e} \), \( p \), \( g_1 \) and \( g_2 \) be defined as above. Given a tuple \((g_1, g_2, g_1^a, g_1^b, g_2^b, g_2^c, \hat{e}(g_1, g_2)^r)\) where \( a, b, c, r \overset{\$}{\leftarrow} \mathbb{Z}_p^* \), we say DBDH assumption holds if for any PPT algorithm \( A \), it decides whether \( \hat{e}(g_1, g_2)^r = \hat{e}(g_1, g_2)^{abc} \) with probability only negligibly larger 1/2.

III. PUBLIC-KEY AUTHENTICATED ENCRYPTION WITH DESIGNATED EQUALITY TEST

A. DEFINITION

A public-key authenticated encryption with designated equality test (PKAE-DET) scheme is defined by the following (probabilistic) polynomial-time algorithms.

- \( pp \leftarrow \text{Setup}(\lambda) \): Given the security parameter \( \lambda \), returns system public parameter \( pp \).
- \((pk_S, sk_S) \leftarrow \text{KeyGen}_{\text{Sender}}(pp)\): Given system public parameter \( pp \), returns the sender’s public/secret key pair \((pk_S, sk_S)\).
- \((pk_R, sk_R) \leftarrow \text{KeyGen}_{\text{Receiver}}(pp)\): Given system public parameter \( pp \), returns the receiver’s public/secret key pair \((pk_R, sk_R)\).
- \((pk_T, sk_T) \leftarrow \text{KeyGen}_{\text{Tester}}(pp)\): Given system public parameter \( pp \), returns the tester’s public/secret key pair \((pk_T, sk_T)\).
- \( C \leftarrow \text{Enc}(sk_S, pk_R, pk_T, M) \): Given a sender’s secret key \( sk_S \), a receiver’s public key \( pk_R \), the tester’s public key \( pk_T \) and a message \( M \), returns a ciphertext \( C \).
- \( M \leftarrow \text{Dec}(sk_R, pk_S, pk_T, C) \): Given a receiver’s secret key \( sk_R \) and a ciphertext \( C \), returns a message \( M \) or ⊥ indicating decryption failure.

- \( Td \leftarrow \text{Aut}(sk_R, pk_S, pk_R) \): Given a receiver’s secret key \( sk_R \), a sender’s public key \( pk_S \) and a targeted receiver’s public key \( pk_R \), returns a trapdoor \( Td \) as the authorization for test.
- \( \text{result} \leftarrow \text{Test}(C, Td, C’, Td’, pk_R, pk_K, sk_T) \): Given two ciphertexts \( C \) and \( C’ \), two trapdoors \( Td \) and \( Td’ \), two receivers’ public key \( pk_R \) and \( pk_K \) and the tester’s secret key \( sk_T \), returns 1 if \( C \) and \( C’ \) contain the same message, and 0 otherwise.

To prevent outside adversaries, it is necessary to designate the tester during the generation of a ciphertext. In addition, in applications like DRG, we need to prevent an inside adversary from generating ciphertexts on behalf of the sender, or comparing the legitimate receiver’s ciphertexts with others if it is not authorized to do so. Therefore, the encryption algorithm of PKAE-DET takes the tester’s public key as input, while the trapdoor generation algorithm takes the sender’s public key and the public key of another receiver for ciphertext comparison as input. That is, the receiver needs to specify whose ciphertexts the tester could compare its ciphertexts with.

B. SECURITY MODELS

Game I: One-wayness under chosen ciphertext attacks (OW-CCA) With authorization from the receiver, the adversary \( A \) (which could be the tester) should not be able to recover the message from a given ciphertext. Consider the following game.

1) Setup: The challenger \( C \) generates system public parameter \( pp \), \( n_S \) key pairs of senders \((pk_S, sk_S)\), \( 1 \leq i \leq n_S \), \( n_R \) key pairs of receivers \((pk_R, sk_R)\), \( 1 \leq j \leq n_R \), and a key pair \((pk_T, sk_T)\) of the tester. It sends \( pp \), \((pk_S), (pk_R)\) and \((pk_T, sk_T)\) to \( A \).

2) Phase 1: \( A \) is allowed to issue the following queries for polynomially many times:

- Key Query\((i)\): Given an index \( i \) of a sender or receiver, the oracle returns the secret key \( sk_i \).
- Decryption Query\((i, j, C)\): Given index \( i \) of a sender, \( j \) of a receiver and a ciphertext \( C \), the oracle returns the corresponding message \( M \).
- Authorization Query\((i, j, k)\): Given index \( i \) of a sender, \( j \) of a receiver and \( k \) of another receiver aiming to compare with, the oracle computes and returns the trapdoor \( Td \).

3) Challenge: \( A \) specifies a challenge sender \( S^* \) and a challenge receiver \( R^* \). \( C \) randomly chooses a message \( M^* \), computes \( C^* \leftarrow \text{Enc}(sk_{S^*}, pk_{R^*}, pk_T, M^*) \) and returns \( C^* \) to \( A \).

4) Phase 2: \( A \) continues to issue queries as in Phase 1.

5) Guess: \( A \) outputs \( M^* \) and wins the game if \( M^* = M^* \) and

a) \( R^* \) was not issued as a key query;

b) \((-, R^*, C^*) \) was not issued as a decryption query;

c) \((S^*, R^*, \cdot) \) was not issued as an authorization query.
The advantage of \( \mathcal{A} \) in the game above is defined to be its probability of winning the game, e.g. \( \text{Adv}_A^{\text{OW-CCA}}(1^\lambda) = \text{Pr}[M' = M^{\ast}] \). We have the following definition.

**Definition 1 (OW-CCA Security):** A PKAE-DET scheme satisfies OW-CCA security if for any PPT adversary \( \mathcal{A} \), its advantage \( \text{Adv}_A^{\text{OW-CCA}}(1^\lambda) \) is negligible.

**Game II: Indistinguishability under chosen ciphertext attacks (IND-CCA):** Without authorization from the receiver, the adversary \( \mathcal{A} \) (which could be the tester) should not be able to obtain even one-bit information about the message from a ciphertext. Consider the following game.

1) **Setup:** Same as that in Game I.
2) **Phase 1:** \( \mathcal{A} \) is allowed to issue key queries and decryption queries as in Game I for polynomially many times.
3) **Challenge:** \( \mathcal{A} \) specifies a challenge sender \( S^\ast \), a challenge receiver \( R^\ast \), and two challenge message \( M^\ast_0 \) and \( M^\ast_1 \) of the same length. \( C \) randomly selects a bit \( b \in \{0, 1\} \), computes \( C^\ast \leftarrow \text{Enc}(sk_{S^\ast}, pk_{R^\ast}, pk_{T}, M^\ast_b) \) and returns \( C^\ast \) to \( \mathcal{A} \).
4) **Phase 2:** Same as that in Phase 1.
5) **Guess:** \( \mathcal{A} \) outputs a bit \( b' \), and wins the game if \( b' = b \) and
a) \( R^\ast \) was not issued as a key query;
b) \( (\cdot, R^\ast, \cdot) \) was not issued as a decryption query;
c) neither \( (\cdot, \cdot, \cdot, \cdot, \cdot, R^\ast, \cdot) \) nor \( (\cdot, \cdot, \cdot, \cdot, \cdot, R^\ast) \) was issued as a test query.

The advantage of \( \mathcal{A} \) in the game above is defined to be \( \text{Adv}_A^{\text{DT-CCA}}(1^\lambda) = |\text{Pr}[b' = b] - \frac{1}{2}| \). We have the following definition.

**Definition 3 (DT-CCA Security):** A PKAE-DET scheme satisfies DT-CCA security if for any PPT adversary \( \mathcal{A} \), its advantage \( \text{Adv}_A^{\text{DT-CCA}}(1^\lambda) \) is negligible.

**IV. CONSTRUCTION**

In this part we present our PKAE-DET scheme. It works as below.

1) **Setup(1^\lambda):** The algorithm generates bilinear group parameters \( (G_1, G_2, G_T, \hat{e}, p) \), chooses two random generators \( g \in G_1, h \in G_2 \), and selects \( f \stackrel{\$}{\leftarrow} G_2 \). It then selects hash functions \( H_1: \{0, 1\}^\ast \rightarrow \mathbb{Z}_p^\ast, H_2: \{0, 1\}^\ast \rightarrow \{0, 1\}^{\ell + \ell_p}, H_3: \{0, 1\}^\ast \rightarrow \{0, 1\}^{l_m} \), where \( l_m \) is maximum length of a message, \( \ell_p \) and \( \ell_T \) are the representation length of an element of group \( \mathbb{Z}_p^\ast \) and \( G_1 \), respectively. It publishes system public parameter \( pp = (G_1, G_2, G_T, p, g, h, f, \hat{e}, H_1, H_2, H_3) \).
2) **KeyGen_{Sender}(pp):** The algorithm selects \( \beta \stackrel{\$}{\leftarrow} \mathbb{Z}_p^\ast \) and outputs \( pk_{S} = g^\beta, sk_{S} = s \).
3) **KeyGen_{Receiver}(pp):** The algorithm selects \( r, w \stackrel{\$}{\leftarrow} \mathbb{Z}_p^\ast \) and outputs \( pk_{R} = (pk_{R,1}^1, pk_{R,2}^1, pk_{R,3}^1, pk_{R},) = (g^r, g^w, h^w, h^r), sk_{R} = (sk_{R,1}^1, sk_{R,2}^1, sk_{R,3}^1) = (r, w, v) \).
4) **KeyGen_{Tester}(pp):** The algorithm selects \( t \stackrel{\$}{\leftarrow} \mathbb{Z}_p^\ast \) and outputs \( pk_T = (pk_T^1, pk_T^2) = (g^t, h^t), sk_T = t \).
5) **Enc(sk_{S}, pk_{R}, pk_{T}, M):** The algorithm selects \( \alpha, \beta \stackrel{\$}{\leftarrow} \mathbb{Z}_p^\ast \) and generates a ciphertext \( C := (C_1, C_2, C_3, C_4, C_5) \) as follows:

\[
C_1 = g \frac{\beta_{sk_{S}}}{pk_{R}} \cdot (pk_T^1)_{sk_{SK}} \oplus H_3(\hat{e}(pk_T^1, pk_{R}^1)) = g \frac{\beta_{sk_{S}}}{pk_{R}} \cdot g^{\cdot \cdot \cdot} \oplus H_3(\hat{e}(g^t, h^t)),
\]

\[
C_2 = (pk_T^2)^t \cdot h^{-\alpha}, \quad C_3 = (pk_{R,3})^\alpha = h^{\cdot \cdot \cdot},
\]

\[
C_4 = (M || |\alpha| \oplus H_2(C_1 || C_2 || C_3 || C_4 || \hat{e}(pk_{R,3}, f)^{\cdot}) = (M || |\alpha| \oplus \hat{H}_2(C_1 || C_2 || C_3 || C_5 || \hat{e}(g^w, f)^{\cdot}, ) \oplus g^\beta.
\]

\[
C_5 = g^\beta.
\]

6) **Dec(sk_{R}, pk_{S}, pk_{R}, pk_{T}, C):** The algorithm computes

\[
(M || |\alpha| = C_1 \oplus \hat{H}_2(C_1 || C_2 || C_3 || C_4 || \hat{e}(C_5, f)^{\cdot})^\cdot),
\]

\[
= C_1 \oplus \hat{H}_2(C_1 || C_2 || C_3 || C_4 || \hat{e}(g^\beta, f)^{\cdot}),
\]

\[
E = H_3(\hat{e}(C_5, pk_T^2)^{\cdot})^\cdot.
\]

It returns \( M \) if \( (pk_T^2)^t = C_3 \) and

\[
(C_1 \oplus E)^{H_1(M)} = g^\alpha \cdot (pk_{R}^1)^{sk_{R}^1 \cdot H_1(M)},
\]

and \( \bot \) otherwise.
7) **Aut(sk_{R}, pk_{S}, pk_{R}):** The algorithm computes and returns

\[
T_d = (pk_S) \cdot (pk_{R}^2)^{sk_{R}^2} = g^{\cdot \cdot \cdot} \cdot g^{\cdot \cdot \cdot r}.
\]
8) \text{Test}(C, Td, C’, T’d, pk_R, pk_k, sk_T): The test algorithm computes as follows.
\begin{align*}
D_1 &= C_1 \oplus H_3(e(C_5, pk_R^4)), \\
D_1’ &= C_1’ \oplus H_3(e(C_5’, pk_R^4’)), \\
T &= e(D_1/Td, C_2) \cdot e(pk_R, C_3)^{sk_T} = e(g, h)^{\frac{a^T}{\alpha H_3}}, \\
T’ &= e(D_1’/T’d, C_2) \cdot e(pk_R’, C_3)^{sk_T} = e(g, h)^{\frac{a’}{\alpha H_3}}.
\end{align*}
It returns 1 if \( T = T’ \) holds, indicating \( M = M’ \), and 0 otherwise.
Correctness of our scheme could be verified in a straightforward way, so we omit it here.

\section{V. SECURITY ANALYSIS}
In this part we prove the security of our PKAE-DET scheme under the given models via a series of games. Below we use \( S_i \) to denote the event that the adversary wins in Game \( i \).

\subsection{A. OW-CCA SECURITY}
\begin{theorem}[OW-CCA] If BDH assumption holds, our PKAE-DET scheme above is OW-CCA secure in the random oracle model.
\end{theorem}
\begin{proof}
Let \( A \) be a PPT adversary against OW-CCA security of our PKAE-DET scheme. Consider the following games.
\begin{itemize}
\item \textbf{Game 1.0:}
\begin{itemize}
\item \textbf{Setup:} The challenger \( C \) generates system public parameter \( pp = (G_1, G_2, \mathbb{G}_T, p, g, h, f, \hat{e}, H_1, H_2, H_3) \), \( n_5 \) key pairs of senders \( (pk_{S_1}, sk_{S_1}) \), \( n_k \) key pairs of receivers \( (pk_R, sk_R) \), and a key pair \( (pk_T, sk_T) \) of the tester. Then \( C \) sends \( pp, \{pk_{S_1}\}, \{pk_R\} \) and \( \{pk_T, sk_T\} \) to \( A \).
\item \textbf{Phase 1:} \( A \) is allowed to issue the following queries for polynomially many times:
\begin{itemize}
\item \( H_1 \) Hash Query(\( M \)): Given a message \( M \), if \( M \) has not been queried, the oracle returns a random element \( H_1(M) = h_1 \leftarrow \mathbb{Z}_p^* \), and stores \( (M, H_1(M)) \) in the List \( L_1 \) which is initially empty. Otherwise, it returns the corresponding answer from List \( L_1 \).
\item \( H_2 \) Hash Query(\( T \)): Given an input \( T \), if \( T \) has not been queried, the oracle returns a random element \( H_2(T) = h_2 \leftarrow \{0,1\}^{\log n}, \) and stores \( (T, H_2(T)) \) in the List \( L_2 \) which is initially empty. Otherwise, it returns the corresponding answer from List \( L_2 \).
\item \( H_3 \) Hash Query(\( T \)): Given an input \( T \), if \( T \) has not been queried, the oracle returns a random element \( H_3(T) = h_3 \leftarrow \{0,1\}^{\log n}, \) and stores \( (T, H_3(T)) \) in the List \( L_3 \) which is initially empty. Otherwise, it returns the corresponding answer from List \( L_3 \).
\item Key Query(\( i \)): Given index \( i \) of a sender (resp. receiver), the oracle returns the secret key \( sk_{S_i} \) (resp. \( sk_R \)).
\item Decryption Query(\( i, j, C \)): Given index \( i \) of a sender, \( j \) of a receiver and a ciphertext \( C \), the oracle runs \( \text{Dec}(sk_j, pk_i, pk_j, pk_T, C) \) and returns the corresponding message \( M \) or \( \perp \) indicating query failure.
\end{itemize}
\end{itemize}
\end{itemize}
\item \textbf{Phase 2:} \( A \) continues to issue queries as in Game 1.0.
\item \textbf{Guess:} \( A \) outputs \( M’ \), and wins the game if \( M = M’ \) and
\begin{itemize}
\item \( R^* \) was not issued as a key query;
\item \( (\cdot, R^*, C^*) \) was not issued as a decryption query;
\item \( (S^*, R^*, \cdot) \) was not issued as an authorization query.
\end{itemize}
The advantage of \( A \) in the game is
\begin{align*}
\text{Adv}_{A}^{\text{OW-CCA}}(1^k) &= \Pr[S_{1.0}] - \Pr[M’ = M^*].
\end{align*}
\item \textbf{Game 1.1:} This game is almost the same as Game 1.0, except the decryption oracle.
\begin{itemize}
\item Decryption Query(\( i, j, C \)): Given index \( i \) of a sender, \( j \) of a receiver and a ciphertext \( C \), the oracle retrieves tuples \( (T, H_2(T)) \) from List \( L_2 \) with \( T = (C_1 || C_2 || C_3 || C_4) \).
\end{itemize}
For each tuple it computes \( (M || \alpha) \leftarrow H_2(T) \oplus C_4 \), and returns \( M \) if \( (pk_{T^i})^\alpha = C_3 \) and
\begin{align*}
(C_1 \oplus H_3(e(C_5, pk_T^{2i} \cdot sk^1)(H_2(M))) = g^\alpha \cdot (pk_{T^i})^{sk^1} H_4(M).
\end{align*}
If no tuple satisfies the conditions above, it returns \( \perp \). Obviously, the probability that \( A \) correctly gets the output of \( H_2 \) without asking the oracle is negligible. So we have:
\begin{align*}
\Pr[S_{1.1}] - \Pr[S_{1.0}] \leq \text{negl}(1^k).
\end{align*}
\end{itemize}
\item \textbf{Game 1.2:} It is almost the same as Game 1.1, except the generation of \( C^*_4 \). In this game \( C \) selects a random element \( W_1 \in \mathbb{G}_T \), computes \( C^*_4 \) as
\begin{align*}
C^*_4 = (M^* || \alpha) \oplus H_2(C_1 || C_2 || C_3 || C_4 || W_1),
\end{align*}
and adds \((C_1 || C_2 || C_3 || C_4 || W_1), C_4 \oplus (M^* || \alpha)) \) to List \( L_2 \).
Denote by \( E_1 \) the event that \( A \) queried \( (C_1 || C_2 || C_3 || C_4 || e(g^w, f)^\beta) \) to \( H_2 \) hash oracle. Game 1.2 is identical to Game 1.1 if \( E_1 \) does not occur. So we have
\begin{align*}
\Pr[S_{1.2}] - \Pr[S_{1.1}] \leq \Pr[E_1].
\end{align*}
Due to the intractability of BDH problem, it is hard for \( A \) to compute \( \hat{e}(g^w, f)^\beta \). Therefore, we have that
\[
\Pr[E_1] \leq \text{negl}(1^\lambda).
\]

**Game 1.3:** It is almost the same as Game 1.2, except the generation of \( C_4^* \). Now \( C \) computes \( C_4 \) as
\[
C_4^* = (M^*\|\alpha) \oplus W_2^*.
\]
The only difference is that a random string \( W_2^* \) is used to replace \( H_2(C_1^*\|C_2^*\|C_3^*\|C_5^*\|W_1^*) \) in the computation of \( C_4^* \). Since \( W_2^* \) is random, the output of \( H_2 \) is also random. Hence, the change does not make difference to the advantage of \( A \), and we have that
\[
\Pr[S_{1.3}] = \Pr[S_{1.2}].
\]

**Game 1.4:** It is almost the same as Game 1.3, except the generation of \( C_4^* \). Now \( C \) simply sets \( C_1^* \) as a random element \( W_3^* \) of length \( l_m + l_p \). It is obvious that
\[
\Pr[S_{1.4}] = \Pr[S_{1.3}].
\]

Notice that in this game \( C_2^*, C_3^*, C_4^* \) and \( C_5^* \) are all random in the view of \( A \), and \( C_1^* \) is the only ciphertext component containing \( M^* \). Since \( M^* \) is randomly chosen and \( H_1 \) is modeled as an oracle, the probability that \( A \) successfully recovers \( M^* \) from \( C_1^* \) is thus negligible. That is,
\[
\Pr[S_{1.4}] \leq \text{negl}(1^\lambda).
\]
Combining all the claims above, we have that:
\[
\text{Adv}^{\text{OW-CCA}}_A(1^\lambda) = \Pr[S_{1.0}] \leq \Pr[S_{1.1}] + \text{negl}(1^\lambda) \leq \Pr[S_{1.2}] + \Pr[E_1] + \text{negl}(1^\lambda) = \Pr[S_{1.3}] + \Pr[E_1] + \text{negl}(1^\lambda) = \Pr[S_{1.4}] + \Pr[E_1] + \text{negl}(1^\lambda) \leq \text{negl}(1^\lambda).
\]
This completes the proof.

**B. IND-CCA SECURITY**

**Theorem 2 (IND-CCA):** If BDH and SXDH assumptions hold, Our PKAE-DET scheme above is IND-CCA secure in the random oracle model.

**Proof 2:** Consider the following games, where \( A \) is a PPT adversary against IND-CCA security of our scheme.

**Game 2.0:**
- **Setup:** Same as that in Game 1.0.
- **Phase 1:** \( A \) is allowed to issue \( H_1, H_2, H_3 \) queries, key queries and decryption queries as in Game 1.0 for polynomially many times.
- **Challenge:** \( A \) specifies a challenge sender \( S^* \), a challenge receiver \( R^* \), and two challenge messages \( M_0^* \) and \( M_1^* \) of the same length. \( C \) randomly selects a bit \( b \in \{0, 1\} \), selects \( \alpha, \beta \in R \), computes and returns \( C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*) \) to \( A \), where

\[
C_1^* = g^{\frac{\alpha}{\alpha + \beta}} \cdot (pk_{R_1})^{ks_v} \oplus H_3(\hat{e}(pk_{R_1}^4, pk_{R_2}^4)^\beta),
\]
\[
C_2^* = (pk_{T_1}^2)^\alpha = h^\alpha, \quad C_3^* = (pk_{T_2}^3)^\alpha = h^{w^\alpha},
\]
\[
C_4^* = (M_0^*\|\alpha) \oplus H_2(C_1^*\|C_2^*\|C_3^*\|C_5^*\|\hat{e}(pk_{T_2}^5, f)^\beta),
\]
\[
C_5^* = g^\beta.
\]

- **Phase 2:** \( A \) continues to issue queries as in **Phase 1**.
- **Guess:** \( A \) outputs a bit \( b' \), and wins the game if \( b' = b \) and

1) neither \( R^* \) nor \( S^* \) was issued as a key query;
2) neither \((\cdot, R^*), \cdot \) nor \((\cdot, S^*, \cdot) \) was issued as a decryption query.

The advantage of \( A \) in the game is
\[
\text{Adv}^\text{IND-CCA}_A(1^\lambda) = |\Pr[S_{2.0}] - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}|.
\]
We proceed by modifying Game 2.0 in the same way as in the proof of Theorem 1, and obtain Game 2.1 to Game 2.4.

We have that
\[
\Pr[S_{2.0}] \leq \Pr[S_{2.4}] + \Pr[E_1] + \text{negl}(1^\lambda).
\]

**Game 2.5:** We further modify Game 2.4, and replace \((pk_{R_1}^1)^{ks_v}\) with a random element \( W_4^* \in \mathbb{G}_1 \) in the computation of \( C_1^* \). Now the challenge ciphertext \( C^* \) is computed as follows.
\[
C_1^* = g^{\frac{\alpha}{\alpha + \beta}} \cdot W_4^* \oplus H_3(\hat{e}(g^\gamma, h^\nu)^\beta),
\]
\[
C_2^* = h^\alpha, \quad C_3^* = h^{w^\alpha},
\]
\[
C_4^* = W_4^*, \quad C_5^* = g^\beta.
\]

Given a tuple \((g, g^\alpha, g^\beta, g^{w^\alpha}, W_4^*) \) and \( W_4^* \) are indistinguishable to \( A \) under the SXDH assumption. Therefore, we have
\[
|\Pr[S_{2.5}] - \Pr[S_{2.4}]| \leq \text{negl}(1^\lambda).
\]

**Game 2.6:** We modify the generation of \( C^* \) as follows:
\[
C_1^* = W_1^*, \quad C_2^* = h^\alpha, \quad C_3^* = h^{w^\alpha},
\]
\[
C_4^* = W_4^*, \quad C_5^* = g^\beta.
\]
Since \( W_4^* \) is random, we have that \( C_1^* \) is also random. Therefore, the replacement of \( C_4^* \) with a random element of the same length makes no difference to the adversary’s advantage. That is,
\[
\Pr[S_{2.6}] = \Pr[S_{2.5}].
\]
Notice that in this game the whole challenge ciphertext \( C^* \) is random in the view of \( A \), and does not leak any information about the bit \( b \). Therefore, \( A \) wins in Game 2.6 with probability at most \( 1/2 \). That is,
\[
\Pr[S_{2.6}] = \frac{1}{2}.
\]
Combining the claims above, we have that
\[
\text{Adv}_{A}^{\text{IND-CCA}}(1^k) = |\Pr[S_{2.0}] - \frac{1}{2}|
\]
\[
\leq |\Pr[S_{2.4}] + \Pr[E_1] + \text{negl}(1^k) - \frac{1}{2}|
\]
\[
\leq |\Pr[S_{2.5}] + \Pr[E_1] + \text{negl}(1^k) - \frac{1}{2}|
\]
\[
= |\Pr[S_{2.6}] + \Pr[E_1] + \text{negl}(1^k) - \frac{1}{2}|
\]
\[
= |\Pr[E_1] + \text{negl}(1^k)| \leq \text{negl}(1^k).
\]
This completes the proof.

C. DT-CCA SECURITY

Theorem 3 (DT-CCA): If BDH and DBDH assumption holds, our PKAE-DET scheme above is DT-CCA secure in the random oracle model.

Proof 3: Let \(A\) be a PPT adversary against DT-CCA security of our PKAE-DET scheme. Consider the following games.

Game 3.0:

- **Setup**: \(A\) submits the index \(R^{*}\) of its challenge receiver. The challenger \(C\) generates and sets system public parameter \(pp = (G_1, G_2, G_T, p, g, h, f, \hat{e}, H_1, H_2, H_3)\), \(n_S\) key pairs of senders \((pk_S, sk_S)\), \(n_R\) key pairs of receivers \((pk_R, sk_R)\), and a key pair of the tester \((pk_T, sk_T)\). \(C\) sends \(pp = [pk_S], [pk_R]\) and \(pk_T\) to \(A\).

- **Phase 1**: \(A\) is allowed to issue queries as in Game 1.0 for polynomially many times. Furthermore, it is also allowed to issue the following queries:
  - Test Query \((C, Td, C', Td', i, j, k, l)\): Given ciphertexts \(C, C'\), trapdoors \(Td, Td'\), indices of two senders \(i, j\) and indices of two receivers \(k, l\), the oracle computes \(M = \text{Dec}(sk_k, pk_j, pk_k, pk_t, C, Td)\). \(M' = \text{Dec}(sk_k, pk_j, pk_j, pk_T, C, Td)\), and returns 1 if \(M = M'\), and 0 otherwise.

- **Challenge**: \(A\) specifies a challenge sender \(S^{*}\) and two challenge messages \(M_0^{*}\) and \(M_1^{*}\) of the same length. \(C\) randomly selects a bit \(b \in \{0, 1\}\), selects \(\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p^{*}\), computes and returns \(C^{*} = (C_1^{*}, C_2^{*}, C_3^{*}, C_4^{*}, C_5^{*})\) to \(A\), where
  \[
  C_1^{*} = g^{\frac{1}{h^{\alpha}}} \cdot (pk_1^{a})^{sk_S^{*}} \oplus H_3(\hat{e}(pk_1^{a}, pk_2^{a})^{\beta})
  \]
  \[
  = g^{\frac{1}{h^{\alpha}}} \cdot g^{\hat{e}(g^x, h^y)^{\beta}} \oplus H_3(\hat{e}(g^x, h^y)^{\beta})
  \]
  \[
  = (pk_2^{\alpha})^{sk_T} = h^\alpha, \quad C_2^{*} = (pk_3^{\alpha})^{sk_R} = h^{\omega^\alpha}
  \]
  \[
  C_4^{*} = (M_2^{*})^{\alpha} \oplus H_2(C_1^{*} \| C_2^{*} \| C_3^{*} \| C_5^{*} \| \hat{e}(pk_2^{\alpha}, f)^{\beta})
  \]
  \[
  = (M_2^{*})^{\alpha} \oplus H_2(C_1^{*} \| C_2^{*} \| C_3^{*} \| C_5^{*} \| \hat{e}(g^x, f^{\beta}))
  \]
  \[
  C_5^{*} = g^{\beta}.
  \]

- **Phase 2**: \(A\) continues to issue queries as in **Phase 1**.

- **Guess**: \(A\) outputs a bit \(b'\), and wins the game if \(b' = b\) and
  1) \(R^{*}\) was not issued as a key query;
  2) \((., R^{*}, .)\) was not issued as a decryption query;
  3) neither \((., ., ., ., ., R^{*}, .)\) nor \((., ., ., ., ., R^{*})\) was issued as a test query.

The advantage of \(A\) in the game is
\[
\text{Adv}_{A}^{\text{DT-CCA}}(1^k) = |\Pr[S_{1.0}] - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}|.
\]

**Game 3.1**: This game is the same as Game 3.0, except the decryption oracle.

- Decryption Query \((i, j, C)\): Given index \(i\) of a sender, \(j\) of a receiver, and a ciphertext \(C\), the oracle retrieves tuples \((T, H_2(T))\) from List \(L_2\) with \(T = (C_1 \| C_2 \| C_3 \| C_5)\).
  For each tuple it computes \((M \| \alpha) \leftarrow H_2(T) \oplus C_4\), and returns \(M\) if \((pk_3^{\beta})^\alpha = C_3\) and
  \[
  (C_1 + H_3(\hat{e}(C_5, pk_2^{\alpha})^{sk_T}))^{H_1(M)} = g^\alpha \cdot pk_1^{sk_1} H_1(M).
  \]
  If no tuple satisfies the conditions above, it returns \(-1\).

Obviously, the probability that \(A\) correctly gets the output of \(H_2\) without asking the oracle is negligible. So we have:
\[
|\Pr[S_{3.1}] - |\Pr[S_{3.0}]|| \leq \text{negl}(1^k).
\]

**Game 3.2**: It is the same as Game 3.1, except the generation of \(C_1^{*}\). In this game \(C\) selects a random element \(W_6^{*} \in G_T\), computes \(C_1^{*}\) as
\[
C_1^{*} = g^{\frac{1}{h^{\alpha}}} \cdot g^{\hat{e}(g^x, f^{\beta})} \oplus H_3(W_6^{*}),
\]
and adds \((W_6^{*}, C_1^{*} \oplus (g^{\frac{1}{h^{\alpha}}} \cdot g^{\hat{e}(g^x, f^{\beta}))}\) to List \(L_3\).

Denote by \(E_2\) the event that \(A\) queried \((\hat{e}(pk_1^{\alpha}, pk_2^{\alpha})^{\beta})\) to \(H_3\) hash oracle. Game 3.2 is identical to Game 3.1 if \(E_2\) does not occur. So we have
\[
|\Pr[S_{3.2}] - |\Pr[S_{3.1}]|| \leq |\Pr[E_2]|.
\]
Due to the intractability of DBDH problem, it is hard for \(A\) to compute \(\hat{e}(g^x, f^{\beta})\). Therefore, we have that
\[
|\Pr[E_2]| \leq \text{negl}(1^k).
\]

**Game 3.3**: This game is the same as Game 3.2, except the generation of \(C_1^{*}\):
\[
C_1^{*} = g^{\frac{1}{h^{\alpha}}} \cdot g^{\hat{e}(g^x, f^{\beta})} \oplus W_7^{*}.
\]
A random string \(W_7^{*}\) is used to replace \(H_3(W_6^{*})\) in the generation of \(C_1^{*}\). Since \(W_6^{*}\) is random, the output of \(H_3\) is also random. Therefore, the change does not make difference to the advantage of \(A\). We have that
\[
|\Pr[S_{3.3}] - |\Pr[S_{3.2}]|.
\]

**Game 3.4**: We modify the game again. Now \(C\) simply sets \(C_1^{*}\) as a random element \(W_8^{*}\) of length \(l_S\). So we have
\[
|\Pr[S_{3.4}] - |\Pr[S_{3.3}]|.
\]
Next we show that \(A\) wins in Game 3.4 with probability only negligibly larger 1/2.

**Lemma 1**: \(\Pr[S_{3.4}]\) is negligibly larger than 1/2 if DBDH assumption holds.
Proof of Lemma 1: Assuming that event \(S_{3,4}\) happens with probability \(\varepsilon + \frac{1}{2}\) in Game 3.4. We build an algorithm \(B\) to break DBDH assumption.

Given a DBDH instance \((G_1, G_2, G_T, p, \hat{e} : G_1 \times G_2 \rightarrow G_T, g_1, g_2, g_1^x, g_2^x, Z)\), where \(x, y, z\) are random elements of \(Z_p^*\), and \(z\) is either equal to \(\hat{e}(g_1, g_2)^{x \cdot z}\) (i.e. \(\hat{b} = 0\)) or a random element (i.e. \(\hat{b} = 1\)). \(B\) aims to output a guess \(\hat{b}'\) of the bit \(\hat{b}\). Consider the following game.

- **Setup:** \(A\) submits the index \(R^*\) of its challenge receiver. \(B\) generates and sets system public parameter \(pp = (G_1, G_2, G_T, p, g = g_1, h = g_2, f = g_2^x, \hat{e}, H_1, H_2, H_3)\), \(n_S\) key pairs of senders \((pk_S, sk_S)\), \(n_R\) key pairs of receivers \((pk_R, sk_R)\), and a key pair of the tester \((pk_T, sk_T)\). Especially, \(B\) sets \(pk_{R^*} = (pk_{R^1}, pk_{R^2}, pk_{R^3}, pk_{R^4}) = (g^r, s_y^1, g_y^y, g^y)\), \(sk_{R^*} = (sk_{R^1}, sk_{R^2}) = (r^*, y, v^*)\) while \(y\) is unknown to \(B\). \(B\) sends \(pp, (pk_S), (pk_R)\) and \(pk_T\) to \(A\).

- **Phase 1:** \(A\) is allowed to issue queries as in Game 3.4.

- **Challenge:** \(A\) specifies a challenge sender \(S^*\) and two challenge message \(M^*\) and \(M^*_1\) of the same length. \(B\) randomly selects a bit \(b \in \{0, 1\}\), selects \(\alpha, \beta \sim Z_p^*\) and \(W^*_R \sim \{0, 1\}^k\), computes and returns \(C^* = (C^*_1, C^*_2, C^*_3, C^*_4, C^*_5)\) to \(A\), where

\[
\begin{align*}
C^*_1 &= W^*_R, \quad C^*_2 = h^{\alpha}, \quad C^*_3 = h^{\alpha \cdot \beta}, \\
C^*_4 &= (M^*_b || \alpha) \oplus H_2(C^*_1 || C^*_2 || C^*_3 || C^*_5) || Z, \\
C^*_5 &= g^b = g^1.
\end{align*}
\]

- **Phase 2:** \(A\) continues to issue queries as in **Phase 1**.

- **Guess:** \(A\) outputs a bit \(\hat{b}'\), and wins the game if \(\hat{b}' = b\) and

1) \(R^*\) was not issued as a key query;
2) \((R^*, \cdot, \cdot, \cdot)\) was not issued as a decryption query;
3) neither \((\cdot, \cdot, \cdot, \cdot, R^*)\) nor \((\cdot, \cdot, \cdot, \cdot, \cdot, R^*)\) was issued as a test query.

If \(\hat{b}' = b, B\) returns \(\hat{b} = 0\) representing that \(Z\) is not a random element, and 1 otherwise.

If \(Z = \hat{e}(g_1, g_2)^{x \cdot z} = \hat{e}(pk_{R^*}^2, f)^b\) (i.e. \(\hat{b} = 0\)), we have

\[
C^*_4 = ((M^*_b || \alpha) \oplus H_2(C^*_1 || C^*_2 || C^*_3 || C^*_5) || \hat{e}(pk_{R^*}^2, f)^b).
\]

Hence, \(C^*_4\) is a real ciphertext of \(M^*_b\), and the view of \(A\) is identical to a real attack (according to Game 3.4). Otherwise (i.e. \(\hat{b} = 1\), \(Z\) is a random element. Thus, ciphertext \(C^*\) contains no information about \(M^*_b\), and the probability that \(A\) wins the game is only \(1/2\). The advantage that \(B\) breaks the DBDH assumption is then

\[
\text{Adv}^{\text{DBDH}}_B(1^\lambda) = |\Pr[\hat{b}' = \hat{b}] - \frac{1}{2}| = |\Pr[\hat{b}' = 0 \land \hat{b} = 0] + \Pr[\hat{b}' = 1 \land \hat{b} = 1] - \frac{1}{2}| = 2\cdot|\varepsilon + \frac{1}{2}| - \frac{1}{2} = \varepsilon.
\]

If \(\varepsilon\) is non-negligible, so is \(\text{Adv}^{\text{DBDH}}_B(1^\lambda)\). Therefore, we have

\[
\Pr[S_{3,4}] - \frac{1}{2} \leq \text{negl}(1^\lambda).
\]

This completes the proof of Lemma 1.

Combining the claims above, we have that:

\[
\text{Adv}_{A}^{\text{DT-CCA}}(1^\lambda) = |\Pr[S_{3,0}] - \frac{1}{2}|
\]

\[
\leq |\Pr[S_{3,1}] + \text{negl}(1^\lambda) - \frac{1}{2}|
\]

\[
\leq |\Pr[S_{3,2}] + \text{negl}(1^\lambda) - \frac{1}{2}|
\]

\[
= |\Pr[S_{3,3}] + \text{negl}(1^\lambda) - \frac{1}{2}|
\]

\[
= |\Pr[S_{3,4}] + \text{negl}(1^\lambda) - \frac{1}{2}|
\]

\[
\leq \text{negl}(1^\lambda).
\]

This completes the proof.

VI. PERFORMANCE EVALUATION

In this section, we compare the computation and communication costs our PKEET scheme with some other related schemes. Furthermore, we do experiments to demonstrate the practicality of our scheme.

The second to the sixth rows of Table 1 show the comparison in terms of computational complexity of key generation, encryption, decryption, authorization and test algorithms, respectively. The comparison in terms of communication complexity of public key, secret key, ciphertext and trapdoor is then given in the next four rows, respectively. Security of the schemes is compared in the eleventh to the twelfth rows. The last three rows indicate whether the schemes need dual-server setting, support designated tester and resist OMRA, respectively.

As shown in Table 1, our scheme requires less computational overhead in authorization, but more in key generation, encryption, decryption and test algorithms. However, FG-PKEET+ scheme and TS-CCA-FA scheme are based on the dual-server setting, which inevitably requires two servers to transmit messages to each other during the test process, the cost of which could not be neglected. Therefore, dual-server based PKEET schemes should require more running time in practical applications. It is worth noting that the schemes with group mechanism, e.g. G-IBEET [28], need to create a group for each comparison and set up group administrator, which reduces the operability of the scheme. Although G-IBEET scheme seems to be more computation-ally efficient, users need to encrypt the data multiple times for comparing with different users, which is inconvenient in practice. In addition, these related schemes do not support designated tester, while our scheme does, thus further improving security.

We implemented three PKEET schemes secure against OMRA without group mechanism: FG-PKEET+ scheme [9], TS-CCE-FA scheme [26] and our scheme. The experiments
| Comp | KeyGen | Enc | Aut | Test | PKBEP-FA(1)[14] | FG-PKEET+[9] | TS-CCE-FA(1)[26] | G-IBEET[28] | Ours |
|------|--------|-----|-----|------|-----------------|-----------------|-----------------|--------------|------|
|      | 3Exp   | 6Exp | 5Exp | 2Exp | 3Exp           | 4Exp            | 2Exp           | 2Exp        | 4Exp |
| Test | 2Exp + 2P | 6P |      |      | 12Exp + 4Iinv | 2P              | 2P              | 4Exp + 5P   |      |

- **Comm**: computational complexity; **Comp**: communication complexity; **KeyGen**: key generation algorithm; **Enc**: encryption algorithm; **Dec**: decryption algorithm; **Aut**: authorization algorithm; **Test**: test algorithm; **EXP**: computational cost of an exponentiation, **h**: bilinear pairing and element inversion, respectively; **pk**, **sk**: length of public key, secret key, ciphertext and trapdoor, respectively; **G**, **G1**, **G2**, **G3**, **Zp**: length of an element of G, G1, G2, Zp, respectively; **l_m**, **l_k**: length of a message and the output of the hash function, respectively. **Sec/wa**: security against adversaries with trapdoors; **Sec/woa**: security against adversaries without trapdoors; **DS**: dual-server setting; **DT**: designated testability; **OMRA**: security against OMRA; *: instantiation based on [25].

Figures 3, 4, 5 and 6 show the computational costs of Enc, Dec, Aut and Test algorithms of the three schemes, respectively. (TS-CCE-FA scheme does not have decryption algorithm.) TS-CCE-FA scheme supports three types of authorization. Since the other two types are used for ciphertext-specific comparisons, we chose TS-CCE-FA with type-I authorization for comparison, which is more similar with our scheme. As TS-CCE-FA scheme provides the private key directly as the output of authorization algorithm, requiring almost no time in trapdoor generation, however, it can only achieve IND-CPA security, while our scheme achieves CCA-type security.

Notice that in these figures we only count the computational costs of the algorithms. Communication costs between the two servers in FG-PKEET+ and TS-CCE-FA are not counted. Thus, efficiency of our scheme is comparable to the other two schemes, and even would be more efficient in...
Figure 7. Classification of patients into DRG [35].

The test process if we take into account communication cost, which is usually the most frequently executed algorithm in practice. Furthermore, our scheme achieves security against OMRA without resorting to the dual server setting, thus providing higher security guarantee. Therefore, we believe that our scheme is more appropriate in applications like diagnostic related groups, which is explained in the next section.

VII. APPLICATION

Figure 7 shows the classification process of diagnostic related groups (DRG). All possible principal diagnoses are first divided into 25 mutually exclusive principal diagnosis areas referred to as MDC (Major Diagnostic Categories), and then divided into specific DRG. Each MDC is constructed to correspond to a major organ system (e.g. respiratory system, circulatory system, digestive system) [35].

As the scenario shown in Fig. 1, our PKAE-DET scheme can be used in medical systems such as DRG system. Below is an example.

The hospital initializes the system and generates public parameters pp. The third-party cloud server, acting as the tester, runs KeyGenTester to generate (pkT, skT) and publishes its public key pkT. Each doctor runs KeyGenSender to generate his key pair (pkS, skS). Similarly, each patient runs KeyGenReceiver to generate his own (pkR, skR). After the doctor diagnoses the patient, an electronic medical record $M = \{M_1||M_2||M_3||M_4||\cdots\} = \{\text{Circulatory System||Coronary Bypass||with Cardiac Catheterization||without MCC||\cdots}\}$ is generated, then encrypted with his secret key skS, the patient’s public key pkR and tester’s public key pkT, in the form of $C = \{C_1||C_2||C_3||C_4||\cdots\} = \{\text{Enc}(M_1)||\text{Enc}(M_2)||\text{Enc}(M_3)||\text{Enc}(M_4)||\cdots\}$, and uploaded to the cloud server. Patients download their medical records $C$ from the cloud and decrypt them locally.

The criterions of DRG, including diagnoses, procedures, age, sex, discharge status, and the presence of complications or comorbidities, are encrypted into $C’ = \{C’_1||C’_2||C’_3||C’_4||C’_5||C’_6||C’_7||C’_8||C’_9||C’_{10}||\cdots\} = \{\text{Enc}(\text{Respiratory System})||\text{Enc}(\text{Circulatory System})||\text{Enc}(\text{Digestive System})||\text{Enc}(\text{Coronary Bypass})||\text{Enc}(\text{Cardiac Catheterization})||\text{Enc}(\text{Cardiac Defibrillator Implant})||\text{Enc}(\text{Cardiac Catheterization})||\text{Enc}(\text{Cardiac Catheterization})||\text{Enc}(\text{without Cardiac Catheterization})||\text{Enc}(\text{with MCC})||\text{Enc}(\text{without MCC})||\cdots\}$ (The order is determined by medical institution) and uploaded to cloud server.

Patient and medical institution generate their authorizations $Td$ and $Td’$ for cloud server respectively. With authorization, the cloud server is able to match an encrypted electronic medical record $C$ of the patient with the encrypted criterions $C’$ by running the test algorithm and return the matching results to the medical institution. Specifically, the cloud server returns the criterion index corresponding to the patient’s medical record, such as the indices of $C_2$, $C_3$, $C_4$, and $C_{10}$. According to Table 2, because of the existence of Enc(Circulatory System), the range is limited to MDC 05. The range is further limited to 231-236 under MDC 05 according to the existence of Enc(Coronary Bypass). The existence of Enc(with Cardiac Catheterization) and Enc(without MCC) eventually determine the DRG to be code 234. In the whole process, the cloud server does not know what each index represents.

However, due to the limited amount of plaintext space available for criterions, an authorized cloud server can generate a large number of fake medical records and compare them with a patient’s real medical record to guess which disease the patient has. Considering patients’ privacy, we hope that during the search process, the cloud server cannot obtain any additional information about patients’ medical records. It requires to prevent the cloud server from illegally matching, which is the security against OMRA, as mentioned above. Therefore, our PKAE-DET scheme proposed in this paper is applicable in the DRG application.

VIII. CONCLUSION

In this paper we proposed the notion of public key authenticated encryption with designated equality test (PKAE-DET),

| Code | Title |
|------|-------|
| ... | MDC 04: Respiratory System |
| ... | MDC 05: Circulatory System  |
| 231 | with Ptca with MCC |
| 232 | without MCC |
| 233 | Coronary Bypass with Cardiac Catheterization with MCC |
| 234 | without Cardiac Catheterization with MCC |
| 235 | without MCC |
| ... | ... |

TABLE 2. Part of DRG codes and titles [36].
which aims to resist offline message recovery attacks. We proposed a concrete construction of PKAE-DET to demonstrate its practicability, which do not require the dual server setting nor any group mechanism. Our scheme is shown to be secure under the given models based on some simple mathematical assumptions. Experiments show that our scheme is efficient, and could be deployed in cloud environment in practice. It could be used to support applications like privacy preserving diagnostic related groups.

REFERENCES

[1] V. Casola, A. Castiglione, K. K. Choo, and C. Esposito, “Healthcare-related data in the cloud: Challenges and opportunities,” IEEE Cloud Comput., vol. 3, no. 6, pp. 10–14, Apr. 2016.
[2] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, “HealthDep: An efficient and secure deduplication scheme for cloud-assisted eHealth systems,” IEEE Trans. Ind. Informat., vol. 14, no. 9, pp. 4101–4112, Sep. 2018.
[3] M. M. Wiley, “Diagnosis related groups (DRGs): Measuring hospital case mix,” in Wiley StatsRef: Statistics Reference Online. New York, NY, USA, 2014.
[4] N. Goldfeld, “The evolution of diagnosis-related groups (DRGs): From its beginnings in case-mix and resource use theory, to its implementation for payment and now for its current utilization for quality within and outside the hospital,” Qual. Manage. Healthcare, vol. 19, no. 1, pp. 3–16, 2010.
[5] D. X. Song, D. A. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. IEEE Symp. Secur. Privacy, May 2000, pp. 44–55.
[6] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 3027, C. Cachin and J. Camenisch, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 506–522.
[7] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, “Probabilistic public key encryption with equality test,” in Topics in Cryptology (Lecture Notes in Computer Science), vol. 5985. Berlin, Germany: Springer, 2010, pp. 119–131.
[8] R. B. Fetter and J. L. Freeman, “Diagnosis related groups: Product line management within hospitals,” Acad. Manage. Rev., vol. 11, no. 1, pp. 41–54, 1986.
[9] Q. Tang, “Public key encryption schemes supporting equality test with authorisation of different granularity,” Int. J. Appl. Cryptogr., vol. 2, no. 4, pp. 304–321, 2012.
[10] Q. Tang, “Towards public key encryption scheme supporting equality test with fine-grained authorization,” in Proc. 16th Austral. Conf. Inf. Secur. Privacy (ACISP), vol. 6812. Berlin, Germany: Springer, 2011, pp. 389–406.
[11] Q. Tang, “Public key encryption supporting plaintext equality test and user-specified authorization,” Secur. Commun. Netw., vol. 5, no. 12, pp. 1351–1362, 2012.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] S. Ma, “Identity-based encryption with outsourced equality test in cloud computing,” Inf. Sci., vol. 328, pp. 389–402, Jan. 2016.
[17] 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.
[18] 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.

YUANHAO WANG received the B.S. degree in network engineering from South China Agricultural University, in 2017, where he is currently pursuing the M.S. degree with the College of Mathematics and Informatics. His research interests include information security, searchable encryption, and digital signature.
QIONG HUANG received the B.S. and M.S. degrees from Fudan University, in 2003 and 2006, respectively, and the Ph.D. degree from the City University of Hong Kong, in 2010. He is currently a Professor with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China. He has published more than 100 research articles in international conferences and journals in the area of cryptography and information security. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis. He has served as a programme committee member in many international conferences.

HONGBO LI received the B.S. and M.S. degrees from South China Agricultural University, Guangzhou, China, where he is currently pursuing the Ph.D. degree with the College of Mathematics and Informatics. His research interests include applied cryptography and cloud security.

JIANYE HUANG received the B.S. and M.S. degrees from South China Agricultural University. He is currently pursuing the Ph.D. degree with the School of Computing and Information Technology, University of Wollongong, NSW, Australia. His research interests include applied cryptography and cloud security.

GUOMIN YANG (SM’17) received the bachelor’s, master’s, and Ph.D. degrees in computer science from the City University of Hong Kong, in 2004, 2006, and 2009, respectively. Before joining University of Wollongong (UOW), in 2012, he was a Research Scientist with the Temasek Laboratories, National University of Singapore. He is currently an Associate Professor with the School of Computing and Information Technology, Institute of Cybersecurity and Cryptology (iC2), UOW, Australia. His research interests include applied cryptography and network security. In 2015, he received the Australian Research Council Discovery Early Career Researcher Award.

WILLY SUSILO (SM’01) received the Ph.D. degree in computer science from the University of Wollongong, Australia. He is currently a Professor and the Head of the School of Computing and Information Technology and the Director of the Institute of Cybersecurity and Cryptology, University of Wollongong. He has authored or co-authored over 300 research articles in the area of cybersecurity and cryptology. His main research interests include cybersecurity, cryptography, and information security. His work has been cited over 9000 times in Google Scholar. He has served as a Program Committee Member in dozens of international conferences. He was awarded the prestigious ARC Future Fellow by the Australian Research Council and the Researcher of the Year award, in 2016, by the University of Wollongong. He is the Editor-in-Chief of the Information journal. He is currently serving as an Associate Editor for several international journals, including Computer Standards and Interface (Elsevier) and International Journal of Information Security (Springer).