Identity Based Key-Insulated Encryption with Outsourced Equality Test

Seth Alornyo, Yanan Zhao, Guobin Zhu, and Hu Xiong
(Corresponding author: Yanan Zhao)

School of Information and Software Engineering, University of Electronic Science and Technology of China
Sichuan-Chengdu
(Email: zynbyxz@gmail.com)

(Received Oct. 2, 2018; Revised and Accepted Mar. 23, 2019; First Online June 11, 2019)

Abstract

We firstly combine the concepts of key-insulated encryption (KIE) and identity-based encryption with the equality test (IBE-ET) to obtain identity-based key-insulated encryption with equality test (IB-KIEET). The scheme inherits the advantages of identity-based encryption (IBE), which simplifying certificate management for public key encryption. Furthermore, the key-insulated mechanism was added in our scheme, which perfectly reduced the possibility of key exposure. Our scheme achieves weak indistinquishable identity chosen ciphertext (W-IND-ID-CCA) security in the random oracle model. Meanwhile, it is indicated that our scheme is feasible and practical through the experimental simulation and theoretical analysis.

Keywords: Identity Based Encryption; Key-Insulated; Outsourced Equality Test

1 Introduction

Due to the rapid popularity of cloud computing, storing data in the cloud (such as photos, videos, emails, and instant messages) has become a trend for individuals and organizations [5, 20]. However, the cloud server cannot be fully trusted to ensure the confidentiality of user data uploaded to the cloud [16]. For this reason, user's data should be encrypted before sending it to the cloud server. Public key encryption seems to be suitable for encryption [1]. But it is unrealistic for users to download all the data from the cloud server each time. Therefore, it is desirable to design a scheme that supports the search function stored on the ciphertext in the cloud server without revealing any information related to these ciphertexts.

Boneh *et al.* [3] proposed the first public key encryption using keyword search (PKE-KS). In the PKE-KS scheme, the user can encrypt the keyword and corresponding data under the user's public key, meanwhile, the user creates a target keyword trapdoor by using his/her private key and then uploads it to the cloud server. Nonetheless, the cloud server can only compare keywords with trapdoors under the same public key. This has become the bottleneck for the development of keyword search. To address this problem, Yang *et al.* [28] proposed the concept of public key encryption scheme (PKE-ET) with equality test based on bilinear pairing. Compared to PKE-KS, the equality test in PKE-ET can be performed between two ciphertexts encrypted in the same public key and different public keys.

Following the works of Yang *et al.* [28], some well-designed schemes with equality test have been constructed [11, 15, 21, 26]. Recently, Sha Ma [18] proposed the notion of identity based encryption with outsourced equality test(IBE-ET) in cloud computing. The abovementioned scheme is the first time to integrate identity-based cryptosystem into public key encryption with equality test, thus it inherits the advantages of both primitives. However, the problem caused by key exposure can't be resisted in this scheme. There is no doubt that key exposure will lead to the destructive consequence, for which Dodis *et al.* [6] proposed the primitive of key-insulated. In their scheme, the secret keys consist of two parts which named user secret key and helper key. The user secret key has been constantly changing, so the possibility of key exposure is significantly reduced. Therefore, a scheme need to be devised that satisfies both the equality test and the key-insulated encryption.

1.1 Related Work

1.1.1 Key-insulated Encryption

In order to reduce the damage which is caused by private key-exposure, Dodis *et al.* [6] firstly introduced the key-insulated encryption. Nevertheless, in this scheme, the total time period number should be determined in advance. Since then, many research results, about key-insulated encryption have been put forward. By introducing the concept of proxy re-encryption, Wang *et al.* [22] processed a key-insulated proxy re-encryption scheme (KIPRE). He *et al.* [8] combined key-insulated encryption with certificateless public key encryption (CL-PKE) and present a concrete paradigm which is called certificateless key-insulated encryption scheme (CLKIE). Hanaoka *et al.* [7] combined identity-based encryption with key-insulated encryption and proposed the first identity-based key-insulated encryption scheme. Later, Bellare and Palacio [2] proposed a new key-insulated encryption scheme. In this scheme, the total time period number doesn't need to be given in advance. Benoît *et al.* [13] processed a identity-based key-insulated encryption scheme without random oracles.

1.1.2 Equality Test

Boneh *et al.* [4] proposed the first public key encryption with keyword search (PKE-KS) scheme. In this scheme, user is able to test the equvalance between two ciphertexts which are encrypted with the same public key. Later, some well-designed PKE-KS schemes were put foward [9, 27, 29]. However, it is unable for user to conduct search functionality for ciphertexts under different public keys. In order to solve this problem, Yang *et al.* [28] presented public key encryption with equality test (PKE-ET). This scheme allows user to search the ciphertexts in different public keys. After that a large amount of schemes corresponding to PKE-ET have been put forward [4,14,19,30]. Although PKE-ET has excellent performance, there are still some problem on key certificate management, which seriously constrain the efficient in practice. To solve this problem, Ma [18] combined PKE-ET and (identity-based encryption) IBE [3, 23] and proposed the first identity-based encryption with equality test (IBE-ET). Different from PKE-ET, IBE-ET solved the problem of key certificate management. In recent year, a series of schemes which focus on IBE-ET have been published. Wu *et al.* [24] presented a dual server IBE-ET which can resist the inner keywords guessing attack. Recently, in order to provide a scheme which achieves IND-ID-CCA security, Lee *et al.* [10] proposed a semi-generic construction of IBE-ET. Unfortunately, IBE-ET can not reduce the damage caused by private key-exposure. So far, there has not been any scheme which can solve private key-exposure problem.

1.2 Our Contribution

To resolve these challenges, we propose identity based key-insulated encryption with equality test (IB-KIEET) in this paper. To summarize, our contribution to this paper consist of three points:

1) We first incoporate the idea of identity-based key-insulated encryption into IBE-ET to propose the IB-KIEET scheme. Specifically, IB-KIEET enables the cloud server to conduct an equivalence test on ciphertext. Meanwhile, IB-KIEET can resist private key exposure;

2) Our scheme achieves Weak-IND-ID-CCA (W-IND-ID-CCA) security, which can prevent an insider attack.

3) Finally, we give the experimental simulation and theoretical analysis which can indicate the feasibility and practicability of our scheme.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, our scheme provide some preliminaries for our construction and formulate the notion of IB-KIEET. In Section 3, we proposed our construction of IB-KIEET and prove its security in Section 4. In Section 5, we compare our work with other related works. In Section 6, we conclude our paper.

2 Preliminaries

2.1 Billinear map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Suppose that g is a generator of $\mathbb{G}$. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies the following properties:

1) Bilinearity: For any $g \in \mathbb{G}$, a and b $\in \mathbb{Z}_p$, $e(g^a, g^b) = e(g, g)^{ab}$.

2) Non-degenerate: $e(g, g) \neq 1$.

3) Computable: There is an efficient algorithm to compute $e(g, g)$ for any $g \in \mathbb{G}$.

2.2 Bilinear Diffie-Hellman (BDH) problem

Let $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$ be two groups of prime order p. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathbb{T}$ be an admissible bilinear map and let g be a generator of $\mathbb{G}$.The BDH problem in $\langle p, \mathbb{G}, \mathbb{G}_\mathbb{T}, e\rangle$ is as follows: Given $\langle g, g^a, g^b, g^c\rangle$, for random a,b,c $\in \mathbb{Z}_p^*$, for any randomized algorithm $\mathbb{A}$ computes value $e(g, g)^{abc} \in \mathbb{G}_T$ with advantage:

$$ADV_{\mathbb{A}}^{BDH} Pr[\mathbb{A}(g, g^a, g^b, g^c) = e(g, g)^{abc}]$$

We say that the BDH assumption holds if for any polynomial-time algorithm $\mathbb{A}$, its advantage $Adv_{\mathbb{A}}^{BDH}$ is negligible.

2.3 Definitions

In this section, we give formal definitions of our scheme. A physically secured helper device is employed in our model to help update user secret key at a time i,we assume our helper device is secured. Our scheme achieves weak chosen ciphertext security (*i.e.* W-IND-ID-CCA) under the defined security model.

Identity based key-insulated encryption with outsourced equality test (IB-KIEET): In identity based encryption with equality test against outsider attack scheme, we specify nine algorithms: Setup, Extract, UserKeyGeneration, DeviceKeyUpdate, UserKeyUpdate, Trapdoor,Encrypt, Decrypt, Test, where $\mathbb{M}$ and $\mathbb{C}$ are its plaintext space and ciphertext space, respectively:

1) **Setup**(λ): It takes as input a security parameter λ , total number of time period $T = N$ and returns the public system parameter K and the master key msk.
2) **Extract**(msk,ID): It takes as input, msk, an arbitrary ID$\in$ $\{0,1\}^*$, system parameter K and returns a secret key dk_{ID} to the user with identity ID. This algorithm is also performed by a PKG. After the algorithm is performed, PKG sends to the user with identity ID via a secure channel.
3) **UserKeyGeneration**(K, N, dk_{ID}): The user key generation algorithm takes the received secret key dk_{ID} and the total number of time periods N.The algorithm outputs user's master private key dk^*_{ID} and set user's initial secret key dk^0_{ID}
4) **DeviceKeyUpdate**(i, j, dk^*_{ID}): The physically secure device takes as input indices i, j for the time periods $(1 \le i, j \le N)$ and a master private key dk^*_{ID}.It outputs a partial secret key $dk'^{i,j}_{ID}$.
5) **UserKeyUpdate**$(i, j, dk^i_{ID}, dk'^{i,j}_{ID})$: It takes as input indices i, j, a secret key dk^i_{ID}, and a partial secret key $dk'^{i,j}_{ID}$.It returns the secret key dk^j_{ID} for time period j.
6) **Trapdoor** (msk,ID,): It takes as input msk and an arbitrary $ID \in \{0,1\}^*$ and returns a trapdoor td for that identity.
7) **Encrypt**(K,i,ID,m): It takes as input K, the index i of the current time period N, an identity ID $\in \{0,1\}^*$ and a plaintext $m \in M$, and returns a ciphertext c as $c = (i, c)$, where $c \in C$.
8) **Decryption**(dk^i_{ID}, i, c): It takes a current private secret key dk^i_{ID} and a ciphertext (i, c) as inputs and returns a plaintext $m \in M$ or a symbol $\perp$ if the ciphertext is invalid.
9) **Test**(C_A, C_B)**:** It takes ciphertext C_A and C_B produced by user A and user B respectively. It output 1 if message associated with C_A and C_B are equal. It outputs 0 otherwise.

Correctness: The algorithm must satisfy the following conditions:

1) When dk^i_{ID} is updated secret decryption key generated by the physically secure DeviceKeyUpdate algorithm given ID as the public key, then

$$\forall m \in M : Decrypt(C, dk^i_{ID}) = M,$$

where $C = Encrypt(ID, M)$ and $C = (i, c)$.

2) When td_A and td_B are trapdoors generated by trapdoor algorithm given ID_A and ID_B as the public keys, then

$$\forall M \in M : Test(C_A, td_A, C_B, td_B) = 1,$$

where C_A= Encrypt(ID_A, M) and C_B= Encrypt(ID_B, M).

3) When td_A and td_B are trapdoors generated by trapdoor algorithm given ID_A and ID_B as the public keys, then

$$\forall M, M^{'} \in M \text{ and } M \neq M^{'},$$

$$Pr[Test(C_A, td_A, C_B, td_B) = 1]$$

is negligible where C_A = Encrypt (ID_A, M) and C_B = Encrypt $(ID_B, M^{'})$.

Security Models:

1) **Setup**: The challenger takes a security parameter λ as input and runs the setup algorithm. It gives the system parameters K to the adversary $\mathbb{A}$ and keeps the master key msk by itself.
2) **Phase 1**: Private decryption key queries (ID_a): The challenger runs the Extract algorithm to generate the private decryption key dk^i_a corresponding to the public key ID_a. It sends dk^i_a to $\mathbb{A}$.
3) **Trapdoor queries** ID_a. The challenger runs the above private decryption key queries on ID_a to get $dk_{ID,a}$ and then generates the trapdoor td_a using $dk_{ID,a}$ via Trapdoor algorithm. Finally, it sends (td_a) to $\mathbb{A}$.
4) **Decryption queries** $(ID_a, (i, C))$**:** The challenger runs the Decryption algorithm to decrypt the ciphertext (i, C_a) by running Extract algorithm to obtain the private secret key $dk^i_{ID,a}$ corresponding to the public key ID_a. Finally, it sends the plaintext M_a to $\mathbb{A}$.
5) **Challenge**: $\mathbb{A}$ submits an identity ID_{ch} on which it wishes to be challenged. The only constraint is that ID_{ch} did not appear in private decryption key queries in Phase 1 but ID_{ch} may appear in trapdoor queries in Phase 1 or in decryption query ID_{ch}. The challenger randomly chooses a plaintext m $\in$ M and sets C^*= Encrypt(ID_{ch}, m, tok^*_{ID}). Finally, it sends C^* to $\mathbb{A}$ as its challenge ciphertext.
6) **Phase 2**: Private decryption key queries ID_a where $ID_a \neq ID_{ch}$. The challenger responds in the same way as in Phase 1.
7) **Trapdoor queries** ID_a. The challenger responds in the same way as in phase 1.
8) **Decryption queries** $(ID_a, C_i) \neq (ID_{ch}, C^*)$. The challenger responds in the same way as in Phase 1.

9) **Guess**: $\mathbb{A}$ submits a guess $m^{'} \in$ M.

Definition 1. *The scheme is W-ID-CCA secure if for all W-IND-ID-CCA adversaries,* $\boldsymbol{Adv}^{W-ID-CCA}_{IB-KIEET,A}(K) = Pr[m = m^{'}]$ *is negligible.*

3 Construction

We provide a detailed construction for the IB-KIEET in this section as follows:

1) **Setup**($\mathbf{l}^{\lambda}$,N): Initially, the system takes a security parameter λ, a time period N and returns public system parameters K, the master secret key msk.

   - The system generates two multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ with the same orime order p of λ length bits and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The system selects an arbitrary generator $g \in \mathbb{G}$.
   - The algorithm exploit a keyed permutation $F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ for a positive integers, K=$k(\lambda)$ and n=$n(\lambda)$. Set a random value k_1 from $\{0,1\}$. Generate a MAC scheme MAC = (GSV) and obtain k_2 by running $G(\lambda)$. Set the master token key $MTK = (k_1, k_2)$. We adopted Lee et al.'s work to resist against insider attack.
   - The system chooses three hash functions: $H_1 : \{0,1\}^t \rightarrow Z_p^*, H_2 : \{0,1\}^* \rightarrow \mathbb{G}, H_3 : T \times \mathbb{G} \times \mathbb{G}_T \rightarrow [0,1]^{t+l}$, where l is the length of random numbers and t is the length of messages. The algorithm randomly picks (α, β) and set g_1=g^{α}, g_2=g^{β}.Next,picks random elements $\{g_3, h, h_1, \ldots, h_N\} \in \mathbb{G}$. It publishes public parameter K=$(T, p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, g_3, h_{N-1}$, MAC, $H_1, H_2, H_3)$ and MSK =(α, β).T is referred to as a MAC Tag.

2) **Extract** (K,MSK,ID): For a given string ID $\in \{0,1\}^*$, public parameter K and MSK,the algorithm compute $h_{ID} = H_1(ID) \in \mathbb{G}$, set master decryption key $mdk_{ID} = (h^{\alpha}_{ID}, h^{\beta}_{ID})$ where (α, β) is the master secret key.

3) **UserKeyGen**(K,mdk,ID): On input mdk_{ID}, the algorithm set $dk_{ID} = (h^{\alpha^{(1)}}_{ID}, h^{\alpha^{(2)}}_{ID})$ where $(\alpha^{(1)}, \alpha^{(2)}) \in Z_p^*$ and parse it as $dk_{ID} = (dk^{(1)}_{ID}, dk^{(2)}_{ID})$, chooses a random elements $\eta \in \mathbb{G}$ and set $dk^*_{ID} = (dk^{(1)}_{ID}/\eta, dk^{(2)}_{ID})$, and set user's initial decryption key as $dk^0_{ID} = (\eta, \phi, \phi, \phi)$.

4) **DevKeyUpdate**(i, j, dk^*_{ID}): On input two indices i, j and dk^*_{ID}, parse dk^*_{ID} as $(dk^{*(1)}_{ID}, dk^{*(2)}_{ID})$, choose $t \in \mathbb{Z}_q^*$ and return a partial secret key $dk'^{i,j}_{ID} = (dk^{*(1)}_{ID}.h^t_j, dk^{*(2)}_{ID}, g^t)$.

5) **UserKeyUpdate**(i, j, dk^*_{ID}): The algorithm on input indices i, j, a secret key dk^i_{ID} and a partial secret key $dk'^{i,j}_{ID} = (x, y, z)$ parse $dk^i_{ID} = (dk^{i(1)}_{ID}, dk^{i(2)}_{ID}$ $, dk^{i(3)}_{ID}, dk^{i(4)}_{ID})$. The algorithm output $dk^j_{ID} = (dk^{j(1)}_{ID}, dk^{j(2)}_{ID}, dk^{j(3)}_{ID}, dk^{j(4)}_{ID})$ where $dk^{j(1)}_{ID} = dk^{i(1)}_{ID}$ and $dk^{i(1)}_{ID} = \eta$ for all i. Therefore $dk^{j(2)}_{ID} = dk^{i(1)}_{ID}.x$,$dk^{j(3)}_{ID} = y$,$dk^{j(4)}_{ID}$=z. The algorithm send $(dk^i_{ID}, dk'^{i,j}_{ID})$ via a secure channel to the user. A new secret key computed at a period i is used to decrypt a specific ciphertext corresponding to period i. If $(dk^i_{ID}, dk'^{i,j}_{ID})$ is deleted as a result of the key update, then ciphertext stored on the cloud server at a period i could not be decrypted by the user. Other similar key-insulated schemes deleted previous secret keys when the current key was updated to a new secret key.

6) **Trapdoor** (ID): For a given string ID $\in \{0,1\}^*$ the algorithm computes $h_{ID} = H_1(ID) \in \mathbb{G}$ and set the trapdoor $td_{ID} = h^{\beta}_{ID}$, td_{ID} is the second element of mdk_{ID}.

7) **Encrypt**(K, ID,m): To encrypt m with a public ID, algorithm selects two random numbers $r_1, r_2 \in Z_p^*$.Then it computes:

$$\begin{aligned} C_1 &= g^{r_1}, \\ C_2 &= W^{r_1}.H_2(e(g_2, h_{ID})^{r_1}) \end{aligned}$$

where

$$\begin{aligned} W^{r_1} &= F(k_1, H(m)), \\ C_3 &= g^{r_2} \\ C_4 &= (m \parallel r_1) \oplus H_3(C_1 \parallel C_2 \parallel P \parallel e(g_1, h_{ID})^{r_2}). \end{aligned}$$

Finally it returns $C = (C_1, C_2, C_3, C_4)$, where $P \leftarrow S(k_2, C_3)$ for the signing algorithm S of the employed MAC, the corresponding tag P is used to verify C_3. The function F is assumed to be a strong pseudorandom permutation and the MAC is existentially unforgeable under chosen message attack.

8) **Decrypt**(C, dk_{ID}, tok_{ID}):On input the ciphertext C, updated secret key dk^i_{ID} and a token $tok_{ID} = (k_1, k_2)$, the algorithm computes:

$$\begin{aligned} m' \parallel r' &= C_4 \oplus H_3(C_1 \parallel C_2 \parallel P \parallel e(C_3, dk^i_{ID})), \\ m' \parallel r' &= H_3(e(C_3, dk^i_{ID})). \end{aligned}$$

Given $P \leftarrow S(k_2, C_3)$ where $P = MAC_{k_2}(C_3)$, the algorithm verify:

$$P' = MAC_{k_2}(C_3) \text{ if } P' = P.$$

Then it checks whether $C_1 = g^{r_1'}$ and C_2=$W^{r_1'}.H_2(e(C_1, h^{\beta}_{ID}))$ where $W^{r_1'}$=$F(k_1, H(m'))$. If both holds,the algorithm return m'. Otherwise, return $\perp$.

9) **Test**$(C_A, td_{ID_A}, C_B, td_{ID_A})$: On input a ciphertext C_A, trapdoor td_A and a given senders' ciphertext C_B. The algorithm test whether M_A=M_B by computing:

$$\begin{aligned} T_A &= \frac{C_{2,_A}}{H_2(e(C_{1,_A}, td_{ID,_A}))}, \\ T_B &= \frac{C_{2,_B}}{H_2(e(C_{1,_B}, td_{ID,_B}))} \end{aligned}$$

the algorithm outputs 1 if the equation holds, outputs 0 otherwise.

Correctness: The conditions that satisfies the above definitions are shown below:

1) Assuming a well-formed ciphertext for ID_A and ID_B. Given the following:

$$\begin{aligned} T_A &= \frac{C_{2,_A}}{H_2(C_{1,_A}, td_{ID,_A})}, \\ &= \frac{W_A^{r_{1,A}} .H_2(e(g_A^{r_1}, h_{ID,A}^{\beta})}{H_2(e(g_A^{r_1}, h_{ID,A}^{\beta})}, \\ &= W_A^{r_{1,A}} \\ T_B &= \frac{C_{2,_B}}{H_2(C_{1,_B}, td_{ID,_B})} \\ &= \frac{W_B^{r_{1,B}} .H_2(e(g_B^{r_1}, h_{ID,B}^{\beta})}{H_2(e(g_B^{r_1}, h_{ID,B}^{\beta})} \\ &= W_B^{r_{1,B}} \end{aligned}$$

It output 1 if the following equation holds. Otherwise output 0.

$$e(C_{1,_A}, T_B) = e(C_{1,_A}, T_A).$$

Therefore,

$$\begin{aligned} e(C_{1,_A}, T_B) &= e(g^{r_{1,A}}, W_B^{r_{1,B}}) = e(g, W_B)^{r_{1,A} r_{1,B}} \\ e(C_{1,_B}, T_A) &= e(g^{r_{1,B}}, W_A^{r_{1,A}}) = e(g, W_A)^{r_{1,A} r_{1,B}} \end{aligned}$$

Where $W_A^{r_1} = F(k_1, m_A)$ and $W_B^{r_1} = F(k_1, m_B)$, given token tok_{ID}=k_1, the function outputs M_A and M_B.If W_A=W_B, then $e(C_{1,_A}, T_B) = e(C_{1,_B}, T_A)$. Test $(C_A, td_{ID,_A}, C_B, td_{ID,_B})$ outputs 1.

2) For any $M_A \neq M_B$, Test $(C_A, td_{ID,_A}, C_B, td_{ID,_B})$= 1, this implies that $e(g, W_A)^{r_{1,A}}$= $e(g, W_B)^{r_{1,B}}$. Hence $Pr[e(g, W_A)$=$(g, W_B)] = \frac{1}{P}$. Therefore, we assume that $Pr[Test(C_{_A}, td_{ID,_A}, C_B, td_{ID,_B})$ =1] is negligible.

4 Security Analysis

Theorem 1. *The Above IB-KIEET Scheme is W-IND-ID-CCA Secure in the Random Oracle Model Assuming BDHP is negligible.*

Proof. Let $\mathcal{A}$ be a PPT adversary attacking the W-IND-CCA security of the above scheme. Suppose that $\mathbb{A}$ runs in time T and makes at most q_H hash queries and q_D decryption queries. Let $Adv_A^{W-IND-CCA}(t, q_H, q_D)$ denote the advantage of $\mathbb{A}$ in the W-IND-ID-CCA experiment. The security proof is done through a sequence of games by [28]. The preliminaries of the original game is considered as follows:

Game G_0 $\alpha \leftarrow Z_q^*$, y=g^{α}, T=N, R=$\emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*$=$g^r, V^*$=$m^r$,
W^* =$H(T, U^*, V^*, y^r) \oplus (m \parallel r)$;

$m \leftarrow A^{o_H, o_2}(T, U^*, V^*, W^*)$, where the oracle works as follows:

O_H: On input a triple $(T, U, V, Y) \in G_1^4$,where a same random value is returned, if the same input is asked multiple times, the same answer will be returned.

O_2: On input a ciphertext (T,U,V,W), it returns the decryption algorithm to decrypt it using the secret key α given within a time N.

Let X_o be the event that $m^{'}$=m in Game G_0. However the probability in Game G_0 is Pr $[S_o]$. Hence we modify Game G_0 and obtain the following game.

Game G_1 $\alpha \leftarrow Z_q^*$, y=g^{α}, T=N, R=$\emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*$=$g^r, V^*$=$m^r, R^* \rightarrow [0,1]^{t+i}, W^*$= $H(T, U^*, V^*, y^r) \oplus (m \parallel r), R$ =$R \cup (T, U^*, V^*(U^*)^{\alpha}, R^*)$;

$m \leftarrow A^{O_H, O_2}(y, T, U^*, V^*, W^*)$, where the oracle works as follows:

O_H: On input a triple $(T, U, V, Y) \in G_1^4$ where if there is an entry (T, U, V, Y, h) in the hash table R, h is returned, otherwise a random value h is selected and returned, and (T, U, V, Y, h) is added to R.

O_2: On input a ciphertext (T, U, V, W), a hash query on (T, U, V, U^{α}) is issued. Suppose the answer is $h \in [0,1]^{t+i}$, then $m \parallel r$ is computed as $h \oplus W$,then a validity check on whether U=g^r and V=m^r is performed. If the check fails, $\perp$ is returned: otherwise, m is returned. The event that $Game_1$ occurs is denoted by S_1. However its observed that G_0=G_1, hence we deduce the probability of the random oracle as:

$$Pr[S_1] = Pr[S_0].$$

In the next game, we further modify the simulation game in an indistinguishable way:

Game G_2 $\alpha \leftarrow Z_q^*$, y=g^{α}, T=N, R=$\emptyset$;

$m \leftarrow G_1$, $r \leftarrow Z_p^*$,U^*=$g^r, V^* = m^r, W^* \rightarrow [0,1]^{t+i}, R = R \cup (t, U^*, V^*(U^*)^{\alpha}, W^*)$;

$m \leftarrow A^{O_H,O_2}(y,T,U^*,V^*,W^*)$. The oracle response to queries as follows:

O_H: Game G_2 is identical to Game G_1. However if Adversary queries for $(U*,.,(U*)^\alpha)$, then the game is aborted. Let ε be this event.

O_2: This is also the same as Game G_1, however if Adversary ask for decryption of (U^*,V^*W), where $W^{'} \neq W^*$, $\perp$ is retuned.

Chosen Ciphertext security (CCA) secure is paramount in this game because W^* is a random value in both Games, however the random oracle responds are unique and probabilistic because W^* is dependent on U and V^*. The probability of $\perp$ occurring is negligible.

In the next game, we further modify the simulation game in a time T based indistinguishable way.

Game G_3 $\alpha \leftarrow Z_q^*, y = g^\alpha, T = N, R = \emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*{=}g^r, V^*{=}m^r, W^* \rightarrow [0,1]^{t+i}, R{=}\ R \cup (T,U^*,V^*(U^*)^\alpha,W^*)$;

$m \leftarrow A^{O_H,O_2}(y,T,U^*,V^*,W^*)$:

O_H: Game G_3 is identical to Game G_2 . However if Adversary queries for $(U^*,T,U^*,.,(U^*)^\alpha)$, then the game is aborted. Let ε_1 be this event.

O_2: This is also the same as Game G_2, however if Adversary ask for decryption of (U^*,V^*,T), where $T^{'} \neq T$, $\perp$ is retuned.

The timestamp associated with the ciphertext improve the security of this game. T is a tampstamp value associated with the ciphertext in both Games, however the random oracle responds are unique and probabilistic because decrption queries are dependent on T, U^* and V^*. The probability of $\perp$ occurring is negligible.

The challenge ciphertext generated in this game is identically distributed to that in Game G_2 and G_3 as W^* is a random value in both Game G_2 and Game G_3. The simulation of O_2 is secure since W^* is uniquely determined by U^* and V^* in Game G_2 and U^*, V^*, T in Game G_3. Therefore, if event ε_1 does not occur, Game G_3 is identical to Game G_1. However, we show below that event ε_1 occurs with negligible probability.

We further simulates decryption queries in indistinquishable way from Game G_3. The decryption queries are separated into two types which includes:

Type 1: (T,U,V,U^α) has been queried to O_H before a decryption query (T,U,V,W) is issued. In this case, W is uniquely determined after (T,U,V,U^α) is queried to O_H. So the decryption oracle is simulated perfecly.

Type 2: (U,V,U^α) has never been queried to O_H when a decryption query (U,V,W) is issued. In this case, $\perp$ is returned by the decryption oracle. The simulation fails if (U,V,W) is a valid ciphertext. However, this happens with negligible probability.

□

5 Comparison

In this section, we compare the efficiency of algorithms and time consumption among the proposed scheme, Ma's [18] scheme, which combined the concepts of public key encryption with equality test and identity-based encryption, Wu *et al.*'s [25] scheme,which solved the problem of the insider attack, and Li *et al.*'s [12] scheme,in which a key-insulation cryptosystem was proposed in order to minimize the damage of secret key exposure. The comparison result of efficiency is shown in Table 1, which includes Outsider Attack(OA), Insider Attach(IA), encryption(Enc), decryption(Dec), Test and Security. The above comparison shows that our scheme can resist both OA and IA, whereas others' don't have this ability. In addition, the scheme in [18, 25] as well as our scheme implement chosen ciphertext security, which is stronger than chosen plaintext security achieved in [12].

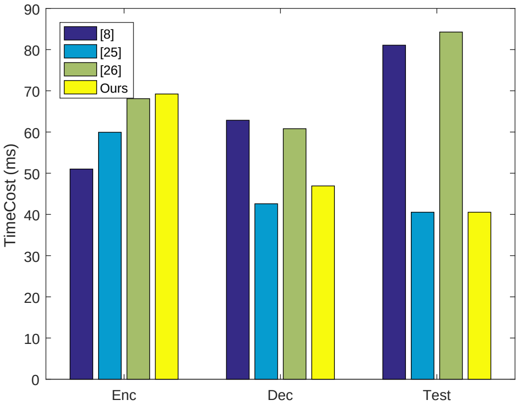

Figure 1: Computation overhead of different schemes

In order to evaluate the computation efficiency of these schemes, the Pairing-Based Cryptography(PBC) Library [17] is used to quantify the time consumption of encryption, decryption and test operations. This experiment is executed on windows 7 OS equipped with an i5-4460 CPU @3.2 GHz and 4G bytes memory. The time consumptions, which are obtained by repeat simulations, are shown in Figure 1. From Figure 1 we can observe that the computation cost of decryption and test of our scheme is comparable with other existing works, whereas our encryption computational cost seems higher. This is forgivable due to the additional computation overheads required to prevent both insider and outsider attacks, which, however, is not the case in other works. In the aspect of the computation cost of decryption and test, our scheme is better than schemes in [12, 25]. Although time consumption of decryption and test operations of our

Table 1: Comparing the efficiency of algorithm of variant PKE-ETs with our scheme

SCHEME	OA	IA	Enc	Dec	Test	Security
[18]	N	N	$4\text{Exp}_1 + 2\text{Exp}_2$	$2\text{P}+2\text{Exp}_1$	4P	OW-ID-CCA
[25]	N	Y	$1\text{P}+3\text{Exp}_1 + 1\text{Exp}_2$	$1\text{P}+2\text{Exp}_1$	2P	W-IND-ID-CCA
[12]	Y	N	$1\text{P}+4\text{Exp}_1 + 1\text{Exp}_2$	3P	$4\text{P}+1\text{Exp}_2$	IND-ID-CPA
Ours	Y	Y	$2\text{P}+2\text{Exp}_1 + 2\text{Exp}_2$	$2\text{P}+2\text{Exp}_1$	2P	W-IND-ID-CCA

legends: In this table, $''Exp_i''$ refers to the exponent computation in group i, $''P''$ refers to the pairing computation, $''OA''$ refers to outsider attack, $''IA''$ refers to insider attack, $''Y''$ refers to 'Yes' as a supportive remark, $''N''$ refers to 'No' as not supportive. W-IND-ID-CCA refers to weak indistinguishable chosen ciphertext attack against identity, OW-ID-CCA refers to one-way chosen ciphertext attack against ientity and IND-ID-CPA refers to indistinguishable chosen plaintext attack against identity.

scheme is slightly high than scheme proposed in [25], it provides additional security for outsider attack.

6 Conclusions

Inspired by the notion of scheme in [18], we put forward identity-based key-insulated encryption with outsourced equality test scheme. In this paper, the mechanism of key-insulated is used to reduce the damage to private key exposure. Besides, our scheme also has the ability to resist insider attack from HBC server, which makes it is practical and suitable in cloud computing. Finally, our scheme security is proved in the random oracle. Theoretical analysis and experiment simulation both demonstrate that our scheme is secure and efficient.

7 Acknowledgements

This work was supported in part by the 13th Five-Year Plan of National Cryptography Development Fund for Cryptographic Theory of China under Grant MMJJ20170204, in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2016J091, the Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems, and in part by the Natural Science Foundation of China under Grants U1401257, 61472064 and 61602096.

References

[1] D. S. AbdElminaam, “Improving the security of cloud computing by building new hybrid cryptography algorithms,” *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] M. Bellare and A. Palacio, ”Protecting against key-exposure: Strongly key-insulated encryption with optimal threshold,” *Applicable Algebra in Engineering, Communication and Computing*, vol. 16, no. 6, pp. 379–396, 2006.

[3] D. Boneh and M. Franklin, ”Identity-based encryption from the weil pairing,” in *Annual International Cryptology Conference*, pp. 213–229, 2001.

[4] D. Boneh, C. G. Di, R. Ostrovsky, and G. Persiano, ”Public key encryption with keyword search,” in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506–522, 2004.

[5] X. F. Cao, H. Li, L. J. Dang, and Y. Lin, ”A two-party privacy preserving set intersection protocol against malicious users in cloud computing,” *Computer Standards & Interfaces*, vol. 54, pp. 41–45, 2017.

[6] Y. Dodis, J. Katz, S. H. Xu, and M. Yung, ”Key-insulated public key cryptosystems,” in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 65–82, 2002.

[7] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai, ”Identity-based hierarchical strongly key-insulated encryption and its application,” in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 495–514, 2005.

[8] L. B. He, C. Yuan, H. Xiong, and Z. G. Qin, ”An efficient and provably secure certificateless key insulated encryption with applications to mobile internet,” *Internationl Journal Network Security*, vol. 19, no. 6, pp. 940–949, 2017.

[9] S. T. Hsu, C. C. Yang, and M. S. Hwang, ”A study of public key encryption with keyword search,” *Internationl Journal Network Security*, vol. 15, no. 2, pp. 71–79, 2013.

[10] H. T. Lee, S. Ling, J. H. Seo, and H. X. Wang, ”Semi-generic construction of public key encryption and identity-based encryption with equality test,” *Information Sciences*, vol. 373, pp. 419–440, 2016.

[11] H. T. Lee, H. X. Wang, and K. Zhang, ”Security analysis and modification of id-based encryption with equality test from acisp 2017,” in *Australasian Conference on Information Security and Privacy*, pp. 780–786, 2018.

[12] J. Li, F. G. Zhang, and T. M. Wang, ”A strong identity based key-insulated cryptosystem,” in *In-*

ternational Conference on Embedded and Ubiquitous Computing, pp. 352–361, 2006.

[13] B. Libert, J. J. Quisquater, and M. Yung, ”Parallel key-insulated public key encryption without random oracles,” in *International Workshop on Public Key Cryptography*, pp. 298–314, 2007.

[14] X. J. Lin, L. Sun, and H. P. Qu, ”Generic construction of public key encryption, identity-based encryption and signcryption with equality test,” *Information Sciences*, vol. 453, pp. 111–126, 2018.

[15] H. Lipmaa, ”Verifiable homomorphic oblivious transfer and private equality test,” in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 416–433, 2003.

[16] L. Liu, Z. Cao, C. Mao, “A note on one outsourcing scheme for big data access control in cloud,” *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.

[17] B. Lynn, ”The stanford pairing based crypto library,” *Privacy Preservation Scheme for Multicast Communications in Smart Buildings of the Smart Grid*, 2013. (`https://blog.csdn.net/vingstar/article/details/17113155`)

[18] S. Ma, ”Identity-based encryption with outsourced equality test in cloud computing,” *Information Sciences*, vol. 328, pp. 389–402, 2016.

[19] H. P. Qu, Z. Yan, J. L. Lin, Q. Zhang, and L. Sun, ”Certificateless public key encryption with equality test,” *Information Sciences*, vol. 462, no. 76–92, 2018.

[20] S. Rezaei, M. Ali Doostari, and M. Bayat, “A lightweight and efficient data sharing scheme for cloud computing,” *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[21] Q. Tang, ”Public key encryption schemes supporting equality test with authorisation of different granularity,” *International Journal of Applied Cryptography*, vol. 2, no. 4, pp. 304–321, 2012.

[22] Y. L. Wang, D. J. Yan, F. G. Li, and H. Xiong, ”A key-insulated proxy re-encryption scheme for data sharing in a cloud environment,” *Internationl Journal Network Security*, vol. 19, no. 4, pp. 623–630, 2017.

[23] B. Waters, ”Efficient identity-based encryption without random oracles,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 114–127, 2005.

[24] L. B. Wu, Y. B. Zhang, K. K. R. Choo, and D. B. He, ”Efficient and secure identity-based encryption scheme with equality test in cloud computing,” *Future Generation Computer Systems*, vol. 73, pp. 22–31, 2017.

[25] T. Wu, S. Ma, Y. Mu, and S. K. Zeng, ”Id-based encryption with equality test against insider attack,” in *Australasian Conference on Information Security and Privacy*, pp. 168–183, 2017.

[26] L. B. Wu, Y. B. Zhang, K. K. R. Choo, and D. B. He, ”Efficient identity-based encryption scheme with equality test in smart city,” *IEEE Transactions on Sustainable Computing*, vol. 3, no. 1, pp. 44–55, 2018.

[27] P. Xu, H. Jin, Q. H. Wu, and W. Wang, ”Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack,” *IEEE Transactions on computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

[28] G. M. Yang, C. H. Tan, Q. Huang, and D. S. Wong, ”Probabilistic public key encryption with equality test,” in *Cryptographers'Track at the RSA Conference*, pp. 119–131, 2010.

[29] Y. Yu, J. B. Ni, H. M. Yang, Y. Mu, and W. Susilo, ”Efficient public key encryption with revocable keyword search,” *Security and Communication Networks*, vol. 7, no. 2, pp. 466–472, 2014.

[30] K. Zhang, J. Chen, H. T. Lee, H. F. Qian, and H. X. Wang, ”Efficient public key encryption with equality test in the standard model,” *Theoretical Computer Science*, vol. 755, pp. 65-80, 2019.

Biography

Seth Alornyo received his Master of Philosophy(M.Phil) degree from Kwame Nkrumah University of Science and Technology in 2014. Currently, he is pursuing his Ph.D. in Software Engineering at University of Electronic Science and Technology of China. His research interests lie in the area of Public Key Encryption and Network Security.

Yanan Zhao is currently pursuing her M.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China. She received her B.S. degree from Jiangxi University of Science and Technology in 2017. Her research interests include identity-based public key cryptography.

Guobin Zhu is an assistant professor at University of Electronic Science and Technology of China (UESTC). He received his Ph.D degree from UESTC in 2014. His research interests include: network security and applied cryptography.

Hu Xiong received his Ph.D. degree from University of Electronic Science and Technology of China (UESTC) in 2009.He is now a professor in the UESTC.His research interests include Cryptography and ad hoc network security