Identity-based Public Key Cryptographic Primitive with Delegated Equality Test Against Insider Attack in Cloud Computing

Seth Alornyo[1,2], Acheampong Edward Mensah[1], and Abraham Opanfo Abbam[1]
(Corresponding author: Seth Alornyo)
School of Information and Software Engineering, University of Electronic Science and Technology of China[1]
4 E 2nd Section, 1st Ring Rd, Jianshe Road, Chenghua, Chengdu, Sichuan, China
Computer Science Department, Koforidua Technical University, Koforidua-Ghana[2]
(Email: sabigseth@outlook.com)
(Received Mar. 3, 2019; Revised and Accepted Oct. 3, 2019; First Online Jan. 23, 2020)

Abstract

The notion of attacks perpetuated by an insider (cloud server) is paramount in this era of cloud computing and data analytics. When a cloud server is delegated with certain responsibilities, it is possible for a cloud server to peddle with users' encrypted data for profit gains. The cloud server takes advantage of it's authorization to launch what we referred to as insider attack. We put forward a new improved scheme on identity-based public key cryptographic primitive which integrate delegated equality test to resist insider attack in cloud computing. Our scheme resist the insider attack perpetuated by the cloud server (insider). We refer to our new scheme as identity-based public key cryptographic primitive with delegated equality test against insider attack in cloud computing (IB-PKC-DETIA). We construct our scheme using a witness based cryptographic primitive with an added pairing operation. Our scheme achieves weak indistinquishable identity chosen ciphertext (W-IND-ID-CCA) security using the random oracle model.

Keywords: Identity Based Encryption; Insider Attack; Witness Based Encryption

1 Introduction

The concept of searchable public key encryption which integrate keyword search (PEKS) was unveiled by [2]. Inspite of this, there has been several works on this cryptographic primitives where a search on ciphertext allows a third party to search over an encrypted data without revealing any information about the ciphertext. A Public key cryptographic primitive with equality test was unveiled by [18] and was used to manage encrypted data for clients. Recently, [11] proposed identity based cryptosystem with integrated equality test (IBE-ET) in cloud computing and it enables the cloud server to verify whether two ciphertext from user A and user B are encryption of the same message. .

However, there has been a recent attack perpetuated by an adversary who is able to launch what is referred to as the insider attack [15]. In this era of cloud computing, equality test function are outsourced to a cloud server to examine whether two ciphertext are encryptions with same message [4]. Such a delegated responsibility to the cloud server gives it the leverage to launch the insider attack on users' ciphertext. This attack when successful enables the cloud server peddle with encrypted data for economic gains. If the cloud server has legitimate access to all users ciphertext and can test their equality, then the cloud server (insider) should be resisted from peddling with users' ciphertext. Recent schemes on insider attack has not been able to fully solve this problem.

Recent works of insider attack by [15] and a security analysis and modification by [19] enables the user to generate a token tok_{ID} to prevent the tester from launching the insider attack. Therefore, their scheme is susceptible to the insider attack because the cloud server was not delegated to perform equality test. When the cloud server is delegated to perform equality test on users ciphertext, it could guess the token tok_{ID} to launch the insider attack. The insider attack resistance scheme proposed by [15] and a security analysis and modification by [19] enable the tester after receiving the secret trapdoor to successfully guess the token tok_{ID} and launch the attack as follows:

1) The cloud server (insider) receives a valid trapdoor T_d and tries to find out the message m and the token from T_d.
2) The cloud server (insider) computes IB-PKC-DETIA ciphertext C of a guessed message $m^{'}$ and a guessed token $tok^{'}_{ID}$.
3) The cloud server (insider) checks whether $Test(C, T_d, tok_{ID})= 1$. The equation holds if a

guess of the message $m^{'}$ and a token $tok_{ID}^{'}$ is successful to the adversary (cloud server). Otherwise, go back to Step 2.

Therefore, if the guess of a message and a token are successful as indicated above, then the cloud server (insider) could launch the insider attack because of the delegated responsibility. Other works of IBE-ET [11] and a security modification by [17] assumed that it's possibe to resist the insider attack. On the contrary, when the cloud server is delegated to conduct equality test, it is possible the adversary can launch the insider attack. Therefore, we propose a new improved scheme to resist insider attack perpetuated by adversary (cloud server) delegated to undertake equality or equivalence test on ciphertext.

1.1 Related Work

Boneh *et al.* [2] first unveiled the primitive of PEKS and was later examined by [11] on their work on off-line keyword guessing attack on recent keyword search schemes. Their work showed that PEKS scheme was vulnerable to the insider attack. A related works on a delegated tester was later unveiled by [14] whereby only the designated tester (server) could perform equality test on the ciphertext. Their scheme concentrated on security of the trapdoor in PEKS and a resistant to insider attack. Chen *et al.* [9,17] also proposed a new general framework for secure public key cryptosystem with keyword search and a dual-server public key primitive with keyword search for secure cloud storage to resist the insider attack so far as there was no collusion by two servers [9]. However, keyword guessing attacks against the insider has being a challenging problem in PEKS until recently, [12] proposed the notion of witness-based searchable cryptographic primitive to resist insider attack in PEKS.

A special type of searchable encryption was unveiled in [18] for a general equality test. However, [11] introduced identity based cryptosystem with equality test (IBE-ET) in cloud computing which integrated the identity-based primitive into public key cryptosystem with equality test [2], it gains the advantages of equality test in [18]. In their construction, search functions were delegated to the service provider.

Existing works on insider attacks mainly focused on PEKS schemes in [8,19] whiles few works on PKEETs extensions [7, 11, 13, 15, 18] and IBE-ET applications in [19] were not resistant to insider attack. To solve the problem of insider attack in IBE-ET, ID-based primitive with equality test against the insider attack was recently put forward by Wu *et al.* [17], the scheme claimed their scheme achieves confidentiality in IBE-ET but the work of Lee *et al.* [9] refuted their claim of weak indistinguishability of IBE-ET. Lee *et al.* [9] modified their security analysis claims to achieve the weak indistinguishability as unveiled in [17]. While in [17], their scheme ensured that the designated users' token tok_{ID} were' changed per a corresponding identity, but in [9] scheme, they ensured that the token was fixed for all group users. Therefore, a fixed token tok_{ID} could successfully enable the cloud server guess a new token $tok_{ID}^{'}$ to launch the insider attack. When a cloud server (insider) is delegated to perform equality test, it is possible for the cloud server to launch the insider attack because a guess of a token is possible. To the best of our knowlege, their scheme cannot resist the insider attack as explained above. A scheme to resist the insider attack with delegated equality test in IBE-ET with the cloud server (insider) authorized to perform equality test is still problem to the research community.

1.2 Our Contribution

Wu *et al.* [17] unveiled a variant to IBE-ET scheme. However, their scheme allowed anyone to perform equality test between two ciphertext hence lack authorization for equality test. The security analysis and modification in [9] did not authorize a third party (cloud server) by generating a trapdoor function for the cloud server to perform equality test. It is not clear whether a computed trapdoor to the cloud server could resist the insider attack as claimed in their security analysis and modification scheme.

To address this problem, we added a pairing operation to the witness cryptographic primitive in [6] to resist insider attack in IBE-ET. Witness based encryption ensure that given a witness relation $R(W, X)$ of an NP language L, an encryption of (m, w) can be tested by a generated trapdoor (m', x). The tester checks if $m' = m$. However, it is difficult to compute w from x under a defined witness relation.

Our scheme achieves Weak-IND-ID-CCA (W-IND-ID-CCA) and a resistant to insider attack. Our scheme achieves a stronger notion of IND-ID-CCA security for IBE-ET using the random oracle model.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, our scheme provide some preliminaries for our construction. In Section 3, our scheme formulate the notion of IB-PKC-DETIA. In Section 4, construction of IB-PKC-DETIA and prove its security in Section 5. In Section 6, we compare our work with other related works. In Section 7, we conclude our paper.

2 Preliminaries

Definition 1. *Billinear map: Let* $\mathbf{G}$ *and* $\mathbf{G}_T$ *be two multiplicative cyclic groups of prime order p. Suppose that* $\mathbf{g}$ *is a generator of* $\mathbf{G}$. *A bilinear map* $\mathbf{e} : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G}_T$ *satisfies the following properties:*

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

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

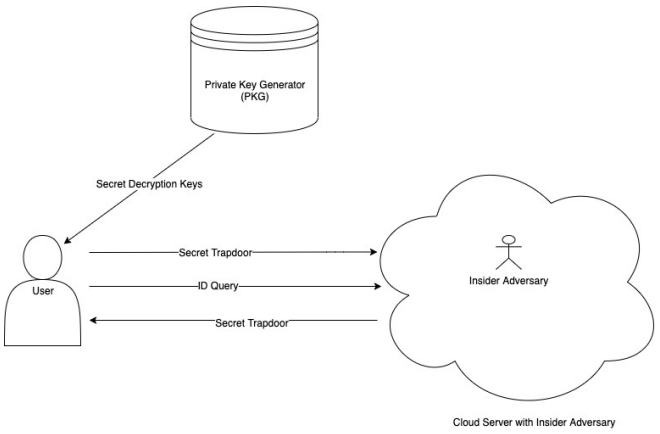

Figure 1: System model for IB-PKC-DETIA

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

Definition 2. *Bilinear Diffie-Hellman (BDH) problem: Let* $\mathbf{G}$,$\mathbf{G_T}$ *be two groups of prime order p. Let* $\mathbf{e} : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G_T}$ *be an admissible bilinear map and let* $\mathbf{g}$ *be a generator of* $\mathbf{G}$. *The BDH problem in* $\langle \mathbf{p}, \mathbf{G}, \mathbf{G_T}, \mathbf{e} \rangle$ *is as follows: Given* $\langle \mathbf{g}, \mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^c \rangle$, *for random a,b,c* $\in \mathbf{Z}_p^*$, *for any randomized algorithm* $\mathbf{A}$ *computes value* $\mathbf{e}(\mathbf{g},\mathbf{g})^{abc} \in \mathbf{G}_T$ *with advantage:*

$ADV_{\mathbf{A}}^{BDH} Pr[\mathbf{A}(\mathbf{g}, \mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^c) = \mathbf{e}(\mathbf{g},\mathbf{g})^{abc}]$. The BDH assumption holds if for any polynomial-time algorithm $\mathbf{A}$, it's advantage $Adv_{\mathbf{A}}^{BDH}$ is negligible.

Definition 3. *Witness Relation: Given a witness relation* $R(W, X)$ *on an NP language L [3], a randomly chosen* $w \in W$ *generates an instance* $x \in X$ *defined over the relation R. For any polynomial algorithm:* $A_{IB-PKC-DETIA}, Pr[A_{IB-PKC-DETIA}(k, w) = x] = 1$, *and* $A_{IB-PKC-DETIA}, Pr[A_{IB-PKC-DETIA}(k, x) = w] < \varepsilon(k)$, *where k is a security parameter and* ε *a negligible functionon on k.*

Our model has four roles which includes: users, PKG, cloud server and with adversary (see Figure 1). Users stores their encrypted sensitive data in the cloud. The cloud server with adversary is resisted from peddling with users encrypted sensitive data for economic gains. In this section, we give formal definitions of our scheme. We employ a witness based cryptographic primitive to resist the insider attack in IBE-ET. Our scheme achieves weak chosen ciphertext security (i.e. W-IND-ID-CCA) under the defined security model.

In identity-based public key cryptographic primitive with delegated equality test against insider attack scheme, we specify seven algorithms: $Setup, Extract, WBInstGen,$ $Trapdoor, WBEncrypt, WBDecrypt, Test$, where $\mathbf{M}$ and $\mathbf{C}$ are its plaintext space and ciphertext space, respectively:

1) **Setup**: It takes as input security parameter k and returns the public key K and msk.

2) **Extract**: It takes as imput msk, an arbitrary $ID \in \{0,1\}^*$ and returns a decryption key dk for that identity.

3) **WBInstGen**: It takes as input the security parameter k, an arbitrary $ID \in \{0,1\}^*$ and returns a private witness key $w \in W$ for that identity w_{ID}, where $WInsGen(w) = x$ and $x \in X$ where (w, x) satisfies the witness relation R.

4) **Trapdoor**: It takes as input decryption key dk, an arbitrary $ID \in \{0,1\}^*$, an instance $x \in X$ and returns a trapdoor td for that identity.

5) **WBEncrypt**: It takes as input an identity $ID \in \{0,1\}^*$, a plaintext $m \in M$ with a random chosen witness $w \in W$ and outputs a ciphertext $C = (x, c)$ where $x \in X$ from a generated witness $WInsGen(w) = x$, and (w,x) satisfies the witness relation R.

6) **WBDecrypt**: The algorithm takes as input the ciphertext $c \in C$, a private decrption key dk and a witness $w \in W$ and returns a plaintext $m \in M$, if and only if C is a valid ciphertext with the ID and a witness $w \in W$.

7) **Test**: It takes as input a ciphertext $C_A \in C$ of a receiver with ID_A, a trapdoor td_A for the receiver with ID_A, a ciphertext $C_B \in C$ of a receiver with ID_B and trapdoor td_B for the receiver with ID_B, and returns 1 if C_A and C_B contains the same message. Otherwise return $\perp$.

3 Security Model

Definition 4. *(Weak-IND-ID-CCA). We let* $\sqcup = (Setup, Extract, WBInstGen, Trapdoor, WBEncrypt, WBDecrypt, Test)$ *be the same scheme and a polynomial time algorithm A.*

1) **Setup**: The challenger runs the security parameter on input k and derives K and randomly takes a witness $w \in W$ and generates an instance $x \in X$ of a witness relation $R(W, X)$ defined on an NP language L. It gives the relation R to the adversary.

2) **Phase 1**: The adversary issues query N_1, N_2, $\cdots$, N_m. Each query is of the form:

   - Query (ID_i): The challenger run $H(.)$ to generate dk_i corresponding to the public identity (ID_i). It sends dk_i to A.

   - Trapdoor (ID_i): The challenger runs the private decryption on $WInsGen$ using a randomly chosen witness $w \in W$ of the relation R(W,X)

on an NP language L. The algorithm generates an instance x and compute a trapdoor td_i using dk_i via trapdoor algorithm. Finally, it sends td_i to A.

- Decryption queries (ID_i, C_i, w): The challenger runs the decryption algorithm to decrypt the ciphertext C_i by running the extract algorithm to obtain dk_i corresponding to the public key (ID_i). Finally, it sends the plaintext M_i to A.

3) **Challenge**: After Phase 1 is over, A submits two equal-length message (m_0, m_1) and ID^* to be challenged by the challenger. However, both (m_0, m_1) are not issued in the encryption query and ID^* is also not in the extract query in Phase 1. The challenger randomly picks $b \in \{0,1\}$ and respond with $C^* \leftarrow Enc(M_b, ID^*, w^*)$. The algorithm generates a challenge trapdoor $td^* = (ID^*, x^*)$ by runing the trapdoor $td^* \leftarrow td(dk, M_b, x^*)$ algorithm and returns td^* to A.

4) **Phase 2**: The adversary issues query N_1, N_2, $\cdots$, N_m. Each query is of the form:

- Query (ID_i). The challenger responds as in Phase 1, since $ID_i \neq ID^*$.
- Trapdoor query (ID_i). Where $x \neq x^*$. The challenger respond in the same way as in Phase 1.
- Decryption Query(ID_i, C_i). Where $(ID_i, C_i) \neq (ID^*, C^*)$, the challenger respond in the same way as in Phase 1.

5) Output: A submits a guess $b^{'}$ on b. If $b^{'} = b$, we say A wins the game.

We define A's advantage on breaking the scheme as $Adv_{IB-PKC-DETIA}(k) = Pr[b^{'} = b] - \frac{1}{2}$ is negligible.

In the Weak-IND-ID-CCA (W-IND-ID-CCA) model, the adversary has access to ciphertexts but cannot compute the witness $w \in W$ from $x \in X$ of a relation $R(W, X)$ over NP language L.

4 Construction

Our scheme aims to resist an attack continuum perpetuated by a cloud server delegated to perform equality test. The cloud server (insider) is considered as an adversary A who is authorized only to perform equlaity test but should not be able to peddle with user's ciphertext. However, only the authorized cloud server could perform equality test.

Definition 5. *A witness relation $R(X, W)$ on an NP language L consist of the following polynomial-time algorithm. Given a randomly chosen $w \in W$ over a witness relation $R(w, x)$=1 defined on an NP language L if for any polynomial algorithm: $A_{IB-PKC-DETIA}, Pr[A_{IB-PKC-DETIA}(k, w) = x] = 1$, and $A_{IB-PKC-DETIA}$, $Pr[A_{IB-PKC-DETIA}(k, x) = w] < \varepsilon(k)$, where k is a security parameter and ε a negligible functionon on k.*

1) **Setup**: The system takes a security parameter k and returns the public parameter K and master secret key msk.

- The algorithm generates the pairing parameters $\mathbf{G}$ and $\mathbf{G}_T$ of prime order p and an admissible bilinear map $\mathbf{e} : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G}_T$. Choose a random generator $\mathbf{g} \in G$.
- The system choose cryptographic hash functions: $H : \{0,1\}^* \rightarrow \mathbf{G}$, $H_1 : \mathbf{G}_T \rightarrow \mathbf{G}$, $H_2 : R \times \mathbf{G}_T \rightarrow \{0,1\}^{\tau_1+\tau_2}$, where τ_1 and τ_2 are security parameters. The elements of $\mathbf{G}$ are represented in τ_1 bits and elements of $x \in X$ of a witness relation R are represented in τ_2 bits.
- The algorithm randomly picks a witness $w \in W$ and generate an instance $x \in X$ on NP language L of a relation $R(W, X)$ and set $\mathbf{g}_1 = \mathbf{g}^w$ and $\mathbf{g}_2 = \mathbf{g}^x$. The ciphertext space $C \in (\mathbf{G}^* \times \{0.1\})^{\tau_1+\tau_2}$. The message space is $M \in \mathbf{G}^*$. It publishes the public parameter $K = \{\mathbf{p}, \mathbf{G}, \mathbf{G}_T, \mathbf{R}, \mathbf{e}, \mathbf{g}, \mathbf{g}_1, \mathbf{g}_2, H, H_1, H_2\}$.

2) **WBInstGen**: The algorithm takes as input the secret parameter k and arbitrary $ID \in \{0,1\}^*$ and compute $h_{ID} = H(ID) \in \mathbf{G}$ and randomly choose a witness $w \in W$ and generates a corresponding instance $x \in X$ on a witness relation $R(W, X)$.

3) **Extract**: Given a string $ID \in \{0,1\}^*$, the algorithm computes $h_{ID} = H(ID) \in \mathbf{G}$ and set the private decryption key $dk_{ID} = (h_{ID}^w, h_{ID}^x)$ where (w, x) is the master key corresponding to the relation R.

4) **Trapdoor**: On input a string $ID \in \{0,1\}^*$, the algorithm compute $h_{ID} = H(ID) \in \mathbf{G}$ and set $td_{ID} = (h_{ID}^x)$ where x is an instance of the relation R.

5) **WBEncypt**: The algorithm on input the public parameter K, ID, it computes $h_{ID} = H(ID) \in \mathbf{G}$ and encrypt $M \in \mathbf{G}$ by choosing two random numbers $(r_1, r_2) \in Z_q^*$ with a randomly chosen witness and instance (w, x). The algorithm set the ciphertext $C = (C_1, C_2, C_3, C_4)$ as:

$$\begin{aligned} C_1 &= M^x.H_2(e(h_{ID}, g_2)^{r_1}), \\ C_2 &= g^{r_1} \\ C_3 &= g^{r_2}, \\ C_4 &= (M \parallel w) \oplus H_2(C_1 \parallel C_2 \parallel C_3 \parallel e(h_{ID}, g_1)^{r_2}). \end{aligned}$$

6) **Decrypt**: To decrypt, the algorithm requires an input the ciphertext C, private decryption key $dk_{ID} = (h^w_{ID}, h^x_{ID})$ where $C = (C_1, C_2, C_3, C_4)$ corresponding to the ciphertext encrypted with ID. It computes:

$$(M^{'} \parallel w^{'}) = C_4 \oplus H_2(C_1 \parallel C_2 \parallel C_3 \parallel e(C_3, h^w_{ID})),$$

where $w \in W$ of the witness relation $R(W, X)$ of the NP language L. The algorithm checks whether $C_1 = (M^{'} \mid x^{'})$ and $C_2 = g^{r_1^{'}}$. Hence, if both holds, return $M^{'}$. Otherwise $\perp$.

7) **Test**: The algorithm on input a ciphertext C_A, a trapdoor td_A and a given sender's ciphertext C_B. The algorithm test whether $M^x_A = M^x_B$ by computing:

$$T_A = \frac{C_{1_A}}{H_2(e(C_{1_A}, td_{ID_A}))}$$
$$T_B = \frac{C_{1_B}}{H_2(e(C_{1_B}, td_{ID_B}))}$$

If the above equation holds, the algorithm outputs 1. Otherwise 0. Thus:

$$e(T_B, C_{2_A}) = e(T_A, C_{2_B}).$$

Remark 1. With the work in [17], the token generated was changed per identity in their construction whiles a security analysis and modification in [9] had a fixed token for all group users. We note that since the token was fixed in their construction, the insider attack is paramount in their scheme. A randomly chosen witness w under the relation $R(W, X)$ is considered to be secure and should avoid a reuse. A reuse of a randomly chosen witness will compromise the security of our scheme hence should be discarded immediately the decryption process is completed.

Correctness:
Let $C_A \leftarrow WBE(M_A, ID_A, g_{2_A}, x_A)$ and $C_B \leftarrow WBE(M_B, ID_B, g_{2_B}, x_B)$ generated by user A and user B respectively. Then:

$$C_A = (C_{A_1}, C_{A_2}, C_{A_3}, C_{A_4}).$$

Test algorithm computes results as:

$C_A = \frac{C_{1_A}}{H_2(e(C_{2_A}, td_{ID_A}))}$, $C_B = \frac{C_{1_B}}{H_2(e(C_{2_B}, td_{ID_B}))}$.

$T_A = \frac{M^{x_A}_A . H_2(e(C_{2_A}, td_{ID_A}))}{H_2(e(C_{2_A}, td_{ID_A}))}$, $T_B = \frac{M^{x_B}_B . H_2(e(C_{2_B}, td_{ID_B}))}{H_2(e(C_{2_B}, td_{ID_B}))}$.

$T_A = \frac{M^{x_A}_A . H_2(e(g_{1_A}^{r_{1_A}}, h^{x_A}_{ID_A}))}{H_2(e(g_{1_A}^{r_{1_A}}, h^{x_A}_{ID_A})}$, $T_B = \frac{M^{x_B}_B . H_2(e(g_{1_B}^{r_{1_B}}, h^{x_B}_{ID_B}))}{H_2(e(g_{1_B}^{r_{1_B}}, h^{x_B}_{ID_B})}$.

$T_A = M^{x_A}_A$ and $T_B = M^{x_B}_B$.

Therefore, $M^{x_A}_A = M^{x_B}_B$.

The algorithm output 1 if the following equation holds. Otherwise 0. Hence:

$$e(C_{2_A}, T_B) = e(C_{2_A}, T_B)$$

$$e(C_{2_A}, T_B) = e(g^{r_{1_A}}, M^{x_B}_B) = e(g, M_B)^{r_{1_A} x_B}$$

$$e(C_{2_B}, T_A) = e(g^{r_{1_B}}, M^{x_A}_A) = e(g, M_A)^{r_{1_B} x_A}$$

Remark 1. Given a witness $w \in W$, the user should be able to compute x to recover M on a witness relation R. However, given the instance x of a witness relation R, it is difficult to recover its corresponding witness w. Therefore the cloud server can only perform equality test but cannot generate a new ciphertext.

If $M_A = M_B$, it implies that $e(C_{2_A}, T_B) = e(C_{2_B}, T_A)$. Given the witness relation $R(W, X)$ defined over an NP language L. It means that: $A_{IB-PKC-DETIA}$, $Pr[A_{IB-PKC-DETIA}(k, w) = x] = 1$, and $A_{IB-PKC-DETIA}$, $Pr[A_{IB-PKC-DETIA}(k, w) = w] < \varepsilon(k)$, where k is a security parameter and ε is a negligible function on k.

5 Security Analysis

We define W-IND-CCA security for IBE-ET via the following game similar in [18].

A probabilistic polynomial time (PPT) adversary A achieves the advantage ε on breaking $\sqcap = (Setup, Extract, WBInstGen, Trapdoor, WBEncrypt, WBDecrypt, Test)$. Given BDH instance, a PPT adversary B takes advantage of A to solve the BDH problem with a probability of $\varepsilon^{'}$.

Suppose B holds a tuple (g,U,V,R) where $a = log_g U$, $b = log_g V$ and $c = log_g R$ are unknown. Given the generator g of G, B is supposed to output $e(g, g)^{abc} \in G_T$. The game between B and A runs as follows:

Setup. B sets $g_1 = g^{a.r_1} = U^{r_1}$, where $r_1 \leftarrow Z^*_q$ and sets trapdoor $td = x \leftarrow X$ from a witness relation $R(W, X)$. B gives g_1 to A.

Phase 1.

1) H Query: A query the random oracle H. A queries ID_i to obtain h_{ID}. B responds with h_{ID}. If ID_i has been in the H table$(ID_i, h^w_{ID_i}, coin)$. Otherwise, for each ID_i, B responds as follows:
   - B tosses a coin with $Pr[coin_i = 0] = \delta$. If $coin_i = 1$, responds to A with $h_{ID} = g^{w_i}$, $w_i \leftarrow W$. Otherwise, B sets $h_{ID} = g^{w_i y} = V^{w_i}$.
   - B responds with h_{ID_i}, then adds $(ID_i, h_{ID}, w_i, coin)$ in the H table which is initially empty.
2) Extract Query: A queries private key of ID_i. B responds as follows:

- B obtains $H(ID_i) = h_{ID}$ in the H table. If $coin_i = 0$, B responds $\perp$ and terminates the game.
- Otherwise, B responds with $dk_{ID_i} = g_1^{w_i} = U^{r_1 \cdot w_i}$, where (w_i, h_{ID_i}) is in the H table.
- B sends dk_{ID_i} to A, then stores (dk_{ID}, ID_i) in the private key list which is initially empty.

3) H_2 Query: A queries $D_i \in R \times G_T \rightarrow \{0,1\}^{\tau_1+\tau_2}$. B responds with $S_i \in H_2(D_i)$ in the H_2 table. Otherwise, for every D_i, B selects a random string $S_i = \{0,1\}^{\tau_1+\tau_2}$ as $H_2(D)$. B responds A with $H_2(D_i)$ and adds (D_i, S_i) in the H_2 table which is initially empty.

4) Trapdoor Query: B runs the private decryption key queries on (ID_i) to obtain $dk_{ID} = (g_1^{w_i}, g_1^{x_i})$ and responds A with $td_{ID_i} = g_2^{x_i}$. td_{ID_i} is the first element of the decryption key.

5) Encryption Query: A queries M_i encrypted with ID_i. B responds as follows:

- B searches the H table and obtain h_{ID} and computes $h_{ID_i} = (h_{ID_i}^{w_i}, h_{ID_i}^{x_i})$ where (w, x) are randomly chosen from the witness relation $R(W, X)$.
- A selects $r_{1_i}, r_{2_i} \leftarrow Z_q^*$ and computes:

$$\begin{aligned} C_{1_i} &= M^{x_i}.H_2(e(h_{ID_i}, g_2)^{r_1}) \\ C_{2_i} &= g^{r_{1_i}} \\ C_{3_i} &= g^{r_{2_i}} \\ D_i &= (C_{1_i}||C_{2_i}||C_{3_i}||e(h_{ID_i}, g_1)^{r_{2_i}}). \end{aligned}$$

- B queries O_{H_2} to obtain $S_i = H_2(D_i)$.
- B computes $C_{4_i} = (M_i||w_i) \oplus S_i$.

Then B responds with $C_i = (C_{1_i}, C_{2_i}, C_{3_i}, C_{4_i})$.

6) Decryption Query: A queries C_i to be decrypted in ID_i. B responds as follows:

- B searches the H table to obtain h_{ID_i}. IF $coin_i = 1$, obtain dk_{ID_i} of ID_i in the private key list to decrypt C_i. Then B computes the bilinear map with dk_{ID_i} as:

$$e(C_{3_i}, h_{ID_i}) = e(g^{r_{2_i}}, g^{w_i}) = e(g, g)^{r_{2_i} w_i}.$$

- B computes $D_i = (C_{1_i}||C_{2_i}||C_{3_i}||e(h_{ID_i}, g_1)^{r_{2_i}})$ and obtains S_i in the H_2 table. B obtains M_i and r_{2_i} by $C_{4_i} \oplus S_i$.

Finally, B computes $C_{1_i}^*, C_{1_i}^*$ with M_i and r_{2_i} decrypted from C_i. If it is a valid ciphertext that $C_{1_i}^* = C_{1_i}$ and $C_{2_i}^* = C_{2_i}$. B responds with M_i. Otherwise $\perp$.

Challenge: Once Phase 1 is over, A outputs two messages m_0, m_1 of equal length and ID^* to be challenged, where both m_0, m_1 are not issued in encryption Query and ID^* is not queried in Extract Query in Phase 1. B responds as follows:

- B searches the H table, if $coin^* = 1$, B responds with $\perp$ and terminates the game, since $h_{ID}^* = g^{w^*}$. It is observed that for a given witness relation $R(W, X)$, it is difficult to compute the corresponding witness w for a given instance x.
- Otherwise, B randomly selects $b \in \{0,1\}$, however, $dk_{ID} = (h_{ID}^{w^*}, h_{ID}^{x^*})$ and calculates:

$$\begin{aligned} C_1^* &= M_b^x.H_2(e(h_{ID}^*, g_2)^{r_1^*}) \\ C_2^* &= g^{r_1^*} \\ C_3^* &= g^{r_2^*} \\ C_4^* &= (M_b||w^*) \oplus S^*, \end{aligned}$$

where $w \in W$ of a witness relation $R(W, X)$, $S^* = H_2(D^*)$ and $D^* = (C_1^*||C_2^*||C_3^*||e(C_3, h_{ID}^w))$, where h_{ID}^w is unknown and B want A to compute. $C^* = (C_1^*||C_2^*||C_3^*||C_4^*$ is a valid ciphertext for M_b.

- B responds A with C^*.

Phase 2:

1) H Query: A queries as in Phase 1.
2) Extract Query: A queries as in Phase 1, but $ID_i \neq ID^*$.
3) H_2: A issues the query as in Phase 1.
4) Trapdoor Query: A queries as in Phase 1, but responds to trapdoor queries the same as in Phase 1. However, the adversary given the witness relation instance $x \in X$, the adversary cannot compute the corresponding witness $w \in W$ of the relation $R(W, X)$ to generate a new ciphertext C^*.
5) Encryption Query: The message $M_i \in \{m_0, m_1\}$, A queries as in Phase 1.
6) Decryption Query: A queries as in Phase 1, except that ciphertext $(C_i, ID_i) \neq (C^*, ID^*)$.

Result: Given a witness relation R, a randomly chosen witness $w \in W$ generates an instance $x \in X$. Given the instance x to compute $td_{ID} = h_{ID}^x$, it is dificult to compute the corresponding witness $w \in W$.

However, A guess $b^{'}$ on b. If $b^{'} \neq b$ and $w^{'} \neq w$, B responds with failure and terminate the game. If $b^{'} = b$ and $w^{'} = w$, then B gets the results of the BDH tuple by guessing the inputs of H_2 query. However, this is not possible under the define witness relation R on NP language L. B aborts the game because, $|Pr[b^{'} = b] - \frac{1}{2}| \geq \frac{\varepsilon}{e(q_{td}+q_e+q_d+1)}$.

Trapdoor Security: We further provide a trapdoor security (TD) experiment to our scheme:

$$Exp_{IB-PKC-DETIA,A}^{W-IND-ID-CCA}(k).$$

Table 1: Efficiency comparisons of PKEETs variant

PKEETs	IA	Enc	Dec	Test	Del	Security
[18]	N	3Exp	3Exp	2P	N/A	OW-CCA
[16]	N	4Exp	2Exp	4P	3Exp	OW/IND-CCA
[15]	N	5Exp	2Exp	4Exp	N/A	OW/IND-CCA
[13]	N	1P+5Exp	1P+4Exp	4P+2Exp	3Exp	OW/IND-CCA
[11]	N	6Exp	2P+2Exp	4P	2Exp	OW/IND-CCA
[17]	Y	1P+3Exp	1P+2Exp	2P	N/A	W-IND-ID-CCA
Ours	Y	2P+3Exp	1P+1Exp	4P	2Exp	W-IND-ID-CCA

Remark: In this table, $''Exp''$ refers to the exponent computation, $''P''$ refers to the pairing computation, $''IA''$ refers to insider attack, $''Y''$ refers to 'Yes' as a supportive remark, $''N''$ refers to 'No' as not supportive and $''Del''$ refers to the delegation, 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.

Given a security parameter k, a witness $w \in W$ and A adversary against trapdoor security (TD). The experiment between the adversary A and the challenger is as follows:

WInsGen Generation Phase: The challenger runs the WInsGen algorithm with a random witness parameter $w \in W$. It generates a corresponding instance $x \in X$. The adversary A computes an instance $x^*(x^* \notin X)$ from a randomly chosen witness w^*. Finally, it gives $td^*_{ID} = h^{x^*}_{ID}$ to A.

Phase 1: A adaptively ask the challenger for the following trapdoor oracle:

1) Trapdoor oracle: On input a message M and instance $x \in X$ where $x \neq x^*$ submitted by A. It output the trapdoor $td_{ID} = h^x_{ID}$ by running the trapdoor algorithm.

2) Challenge Phase: A submit two messages (m_0, m_1) with equal length. The challenger picks $b \leftarrow \{0,1\}$ and generate challenge trapdoor $td^*_{ID} = h^{x^*}_{ID}$ corresponding to the challenge ciphertext $M_b||x \oplus H_2(e(h^x_{ID}, C_2))$ by running the WInsGen algorithm and returns td^*_{ID} to A.

Phase 2:

1) TD_{ID} Query: A continue to ask the oracle for trapdoor queries. Oracle responds as in Phase 1.

2) Output: A output its quess $b^{'}$. The adversary win the game if $b^{'} = b$, which shows that the output of experiment is 1 and 0 otherwise. Adversary A advantage in the above experiment is defined as:

$$\begin{aligned} & Adv^{W-IND-TD}_{IB-PKC-DETIA}(w) \\ = \; & |Pr[Exp^{W-IND-TD}_{IB-PKC-DETIA}] - \frac{1}{2}. \end{aligned}$$

6 Comparison

In this section, we made a comparison (Table 1) on the efficiency of algorithms adopted in our scheme with other PKEET variants. Other PKEET variants (Table 1) achieved a one-way chosen ciphertext attack (OW-CCA), one-way indistinquishable chosen ciphertext attack (OW/IND-CCA) and a weak indistinquishable identity chosen ciphertext attack (W-IND-ID-CCA) security. The extended PKEET schemes cost three to four steps to conduct the equality test including analyzing trapdoor and inverse-computing trapdoor.

The above comparison shows that our scheme can resist insider attack, whereas others do not have such ability except in [17]. Even though Wu *et al.*'s scheme resist insider attack, it does not provide delegation to the cloud server (insider) to perform equality test. It is possible for Wu *et al.*'s scheme to fail the insider attack resistance when the cloud server is delegated to perform equality test. However, our scheme ensures that equality test is delegated to the cloud server and the cloudserver is resisted from launching the insider attack.

The experiment results are shown in Figure 2. This experiment is executed on a desktop computer with an i5-4460 CPU @3.2 GHz and 4gigabyte RAM, running Windows 7, 64 bit system and VC++ 6.0, by using PBC Library [10]. The time consumptions were obtained from a repeated simulations to obtain an objective computational cost comparison (see Figure 2) of ours with Yang *et al.* [18], Tang *et al.* [15,16], Ma *et al.* [11,13], and Wu *et al.* [17]. We assume both schemes were experimented on the same desktop computer.

Obviously, our encryption (Enc) computational cost seems higer than other related schemes. This is due to the extra computational overheads by a generation of an instance from a witness relation to resist insider adversary. Decryptions and test computations were comparable to other schemes (see Figure 2). Although time consumption of encryption (Enc) is slightly higher than in [17] scheme, it provides enhanced security to resist insider attack.

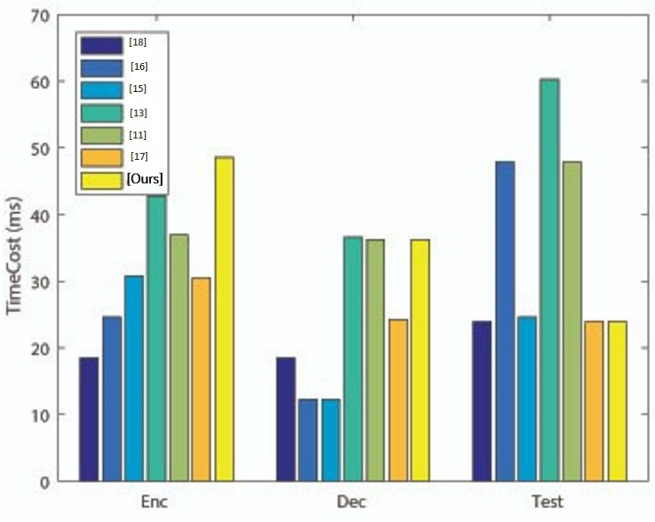

Figure 2: Computational time consumption comparison

7 Conclusion

Our scheme ensures a security improvement in [9,17]. We delegate a cloud server to perform equality test on users ciphertext. Such authorization cause the cloud server to launch the insider attack which our scheme resist such an attack. However, our scheme ensures that even though the cloud server is authorized to perform equality test, it could not launch the insider attack on users cipheretext. Our scheme ensures a resistant to insider attack by the adoption of witness based cryptographic primitive. Our scheme support weak indistinguishable identity chosen ciphertext attack security (W-IND-ID-CCA) with extended trapdoor security (TD).

References

[1] S. Alornyo, M. Asante, X. Hu, and K. K. Mireku, "Encrypted traffic analytic using identity based encryption with equality test for cloud computing," in *IEEE the 7th International Conference on Adaptive Science and Technology (ICAST'18)*, pp. 1-4, 2018.

[2] D. Boneh, G. D. Crescenzo, 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.

[3] J. W. Byun, H. S. Rhee, H. A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Workshop on Secure Data Management*, pp. 75-83, 2006.

[4] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in *Australasian Conference on Information Security and Privacy*, pp. 59-76, 2015.

[5] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789-798, 2016.

[6] S. Garg, C. Gentry, A. Sahai, and B. Waters, "Witness encryption and its applications," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pp. 467-476, 2013.

[7] 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," *The Computer Journal*, vol. 58, no. 10, pp. 2686-2697, 2015.

[8] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing PEKS schemes secure against keyword guessing attacks is possible?," *Computer Communications*, vol. 32, no. 2, pp. 394-396, 2019.

[9] H. T. Lee, H. Wang, and K. Zhang, "Security analysis and modification of ID-based encryption with equality test from ACISP 2017," *Information Security and Privacy*, pp. 780-786, 2018.

[10] B. Lynn, *The Stanford Pairing based Crypto Library.* (`http://crypto.stanford.edu/pbc/`)

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

[12] S. Ma, Y. Mu, W. Susilo, and B. Yang, "Witness-based searchable encryption," *Information Sciences*, vol. 453, pp. 364-378, 2018.

[13] S. Ma, M. Zhang, Q. Huang, and B. Yang, "Public key encryption with delegated equality test in a multi-user setting," *The Computer Journal*, vol. 58, no. 4, pp. 986-1002, 2015.

[14] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *Journal of Systems and Software*, vol. 83, no. 5, pp. 763-771, 2010.

[15] 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.

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

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

[18] G. 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.

[19] W. C. Yau, S. H. Heng, and B. M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *International Conference on Autonomic and Trusted Computing*, pp. 100-105, 2008.

Acknowledgments

We wish to thank the anonymous reviewers for their comments and contributions.

Biography

Seth Alornyo is a lecturer at Koforidua Technical University. He received his Master of Philosophy(M.Phil) degree from Kwame Nkrumah University of Science and Technology in 2014, bachelor of science degree, computer science in 2012 and a higher national diploma in 2008. Currently, he is pursuing his Ph.D. in Software Engineering at University of Electronic Science and Technology of China. His research interests are Cryptography and Network Security. A Member of IEEE.

Acheampong Edward Mensah received his masters degree of Software Engineering from the University of Electronic Science and Technology of China in 2019. His research interests are in the areas of cryptography and information security.

Abraham Opanfo Abbam received his bachelor in Information and Communication Technology degree from University of Education, Winneba. He is currently pursuing software engineering masters degree program at University of Electronic Science and Technology of China. His research interest lies in the area of Deep Learning specifically Natural Language Processing.