Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, *-*
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Parallel Key Insulated ID-Based Public Key Cryptographic Primitive with Outsourced Equality Test

Seth Alornyo[1], Mustapha Adamu Mohammed[1,2], Bright Anibrika Selorm Kodzo[1], Pious Akwasi Sarpong[3], Michael Asante[2]

[1]Koforidua Technical University, Computer Science Department, Koforidua, Ghana
[2]Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
[3]SDA College of Education, Asokore, Koforidua
Email: sabigseth@ktu.edu.gh

How to cite this paper: Author 1, Author 2 and Author 3 (2020) Paper Title. *Journal of Computer and Communications*, **8**, *-*.
https://doi.org/10.4236/***.2020.*****

Received: **** **, ***
Accepted: **** **, ***
Published: **** **, ***

Abstract

Parallel key-insulation allows the use of multiple helper keys to protect private decryption keys during secret decryption key updates. This approach prevents decryption key leakage or exposure in insecure environment. We combined parallel key-insulated encryption (PKIE) with multiple helper keys and identity-based encryption with the equality test (IBE-ET) to obtain parallel key insulated ID-based public key encryption with outsourced equivalent test (PKI-IBPKE-ET). The scheme inherits the advantages of identity-based encryption (IBE), which simplifies certificate management for public key encryption. Furthermore, the parallel key-insulation with multiple helper mechanism was introduced in our scheme, which perfectly reduced the possibility of helper key exposure. Our scheme will enable the protection and periodic update of decryption keys in insecure environment. Our scheme achieves a weak indistinguishable identity chosen ciphertext (W-IND-ID-CCA) security in the random oracle model. Ultimately, it is observed that our scheme is feasible and practical through the experimental simulation and theoretical analysis.

Keywords

Identity-Based Encryption, Equality Test, Parallel Key-Insulated

1. Introduction

Due to the rapid growth of cloud computing [1], storing a user data in the cloud such as photos, moving pictures and other instant electronic messages has attracted the attention of individuals and organizations. However, cloud servers cannot be fully trusted in providing confidentiality and privacy of users data outsourced to the cloud. In this era of traffic analytics and privacy concerns, it is advisable to outsource user's data to the cloud encrypted. Public key cryptosystem has proven to be suitable for encrypting such data. But it may be unrealistic for data owners to access all the data from the cloud via download whenever there is a need to access their files. Therefore, it is desirable to design a scheme that supports the search function on the ciphertext in the cloud server without the need to divulge any information related to these ciphertexts.

Boneh and Franklin *et al.* [2] proposed the first public key cryptographic primitive using keyword search (PKE-KS). In their scheme, the user could encrypt the keyword and corresponding data under a specific user's identity, meanwhile, each user is allowed to create a target keyword trapdoor by using their secret key and then outsource it to the cloud. Nonetheless, outsourced servers can only then do comparison of keywords with trapdoors under similar public key. This has become the bottleneck for the development of keyword search because there is the need to do keyword search with different public keys. To forestall this problem, [3] proposed a public key cryptographic primitive with equality test based on the pairing bilinearity. Compared to PKE-KS, the equality or equivalent test of PKE-ET can be performed among two encrypted data (ciphertexts) with the similar public key, and with different public keys.

Following the construction of Yang *et al.* [3] scheme, some proposed schemes with equivalent test have been constructed [4] [5] [6] [7]. Recently, Ma [8] notion of a cryptographic primitive with equality test (IBE-ET) outsourced to the cloud is the first to blend identity-based cryptographic primitive with public key cryptosystem with equivalent test and this inherited the gains of such schemes. Thus, the problem caused by key exposure could not be avoided in their scheme. Undoubtably, key exposure will lead to a destructive consequence. In view of that, Dodis *et al.* [9] proposed the primitive of key-insulated cryptographic scheme. In their scheme, secret keys consist of two parts namely, secret user key and helper key. The purpose of the secret key adoption is to change frequently so as to prevent the likelihood of key exposure whereas the helper is adopted to help update the secret keys to reduce the exposure of secret decryption keys. Constructing a key-insulated scheme with helper keys that supports public key cryptosystem with equality test is still an open problem. Therefore, a scheme needed to be devised that satisfies both equality test and key-insulation in public key cryptography.

2. Related Work

2.1. Parallel Key-Insulated Cryptosystem

It is of importance to lessen the destructive effect generated by key exposure. Dodis *et al.* [9] first designed the key-insulation cryptosystem. However, in their scheme, the total time period number is determined in advance. Later, [10] proposed new key-insulated crytographic scheme. In their scheme, the total time period number does not need to be given in advance. Since then, many research results about key-insulated encryption have been propounded. By introducing the concept of proxy cryptographic re-encryption scheme, Wang *et al.* [11] constructed a key-insulated proxy cryptographic re-encryption scheme (KIPRE). He *et al.* [12] combined key-insulated encryption with certificateless public key encryption (CL-PKE) and designed a new paradigm of certificateless key-insulated cryptographic encryption scheme (CLKIE). Hanaoka *et al.* [13] combined identity based cryptographic scheme with key-insulation and constructed a novel identity based key-insulation cryptosystem with a single helper. Benot *et al.* [14] also constructed another identity based key-insulation scheme without the adoption of the random oracles.

Introduction of a helper in key-insulated encryption schemes helps to curtail the problem of decryption keys exposure. Thus, temporal secret keys are maintained by users and are refreshed via a mutual interaction between the user and helper. In key-insulated cryptosystems, it is required to often update the keys to reduce the risk of temporal decryption key exposure. This phenomenom requires frequent increase of helper connection during secret key updates in an insecure environment, hence makes the helper key prone to key exposure attacks. To curtail the tendency of helper keys exposure, Hanoaka *et al.* [15] constructed parallel key-insulation encryption (PKIE) to avoid the problem of helper exposure. Their scheme ensured that two distinct helpers alternatively update the secret key. This avoided or curtailed the exposure of a single helper. Therefore, securing the helper has become a major concern in PKIE. Most PKIE schemes [14] [15] [16] [17] adopted the helper approach to avoid the exposure of temporal secret keys and helper keys. Recently, Ren *et al.* [18] proposed multiple helper keys to reduce the risk of using one or two helpers for key updates. A secured helper in key-insulated public key encryption (KIPE) plays a vital role in users secret key updates.

2.2. Equality Test

The first public key cryptograhic scheme with keyword search was announced by Boneh *et al.* [19]. In their construction, users are able to test the equivalance between two encrypted data which are ciphertext with the same public key. Later, some well-designed PKE-KS schemes were constructed [20] [21] with search functions on ciphertexts with different public keys. To solve this problem, [3] constructed encryption scheme with equality test. Their scheme allowed users to search the ciphertexts in different public keys. After that, a large amount of schemes corresponding to PKE-ET have been propounded [19] [22] [23] [24]. Although PKE-ET has excellent performance, there are some inherent problems with key certificate management, that put serious constraints with regards to efficiency and practice. To solve this problem, Ma [18] combined PKE-ET and identity-based scheme (IBE) [2] [25] and reported the first identity-based cryptographic scheme with equality test (IBE-ET). Different from the public key cryptosystem with equality test, identity based cryptosystem solved the problem of key certificate management in public key cryptosystem with equality test. Recently, there have been other applications of identity based cryptographic primitive to detect and prevent malware in encrypted traffic [26]. Also, [27] constructed a dual server identity based cryptosystem which can resist the inner keywords guessing attack so as to prevent an attack on keyword search in public key cryptosystems. In order to provide a scheme that achieves indistiguishable identity chosen ciphertext attack (IND-ID-CCA) security, Lee *et al.* [28] constructed a semi-generic cryptosystem with equality test. Unfortunately, their scheme could not prevent the damage that emanate from private key exposure. So far, there has not been any scheme that can solve private key exposure and helper keys exposure problem in identity (ID) based cryptosystem with equality test.

2.3. Our Contribution

To address these challenges, we propose parallel key insulated ID-based public key encryption with outsourced equality test (PKI-IBPKE-ET). In summary, our contributions to this work consist of three points: 1) We first incoporate the idea of identity-based (ID) parallel key-insulated cryptosystem with multiple helper keys into IBE-ET to construct PKI-IBPKE-ET scheme. Specifically, PKI-IBPKE-ET enables the cloud server to perform equivalence test on ciphertext. Meanwhile, PKI-IBPKE-ET can resist helper key exposure and private decryption key exposure; 2) Our scheme achieves Weak-IND-ID-CCA (W-IND-ID-CCA) security, which also prevents an insider attack [29]; 3) Finally, we give the experimental simulation and theoretical analysis which shows the feasibility and practicability of our novel scheme.

2.4. Paper Organization

The rest of this work is organized as follows; In Section 3, our scheme outlines preliminaries for the construction and the definitions of PKI-IBPKE-ET. In Section 4, the security model is outlined, Section 5 outlines our construction of PKI-IBPKE-ET and proof the security in Section 6. Section 7 compares our work with existing schemes. Section 8 gives a conclusion remark.

3. Scheme Preliminaries

3.1. Bilinear Map

Let G and G_T be two multiplicative cyclic groups of prime order p. We assume that g is a generator of G. A bilinear map $e: G \times G \to G_T$ satisfies the following properties:

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

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

3) Computable: There is an efficient algorithm to compute $e(g,g)$ for any

$g \in G$

3.2. Bilinear Diffie-Hellman (BDH) Problem

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

$$ADV_A^{BDH} = Pr\left[A\left(g, g^a, g^b, g^c\right) = e(g,g)^{abc}\right]. \tag{1}$$

The BDH assumption holds if for all polynomial-time algorithm A, its advantage ADV_A^{BDH} is negligible.

3.3. Definitions

This section gives formal definitions of our proposed scheme. A parallel key-insulated with a multiple helper keys [18] were adopted in our model to do away with the exposure of a single or double helpers and to increase the security of user secret keys. The system model is sketched in **Figure 1**. Our method achieves weak chosen ciphertext security (*i.e.* W-IND-ID-CCA) under a specified security construction model.

In parallel key-insulated ID-based public key encryption with equality test (PKI-IBPKE-ET), we specify nine algorithms: Setup, Extract, UserKeyGeneration, BaseKeyUpdate, UserTempKeyUpdate, PKITrapdoor, PKIEncrypt, PKIDecrypt, Test, M and CT are the plaintext space and ciphertext space, respectively;

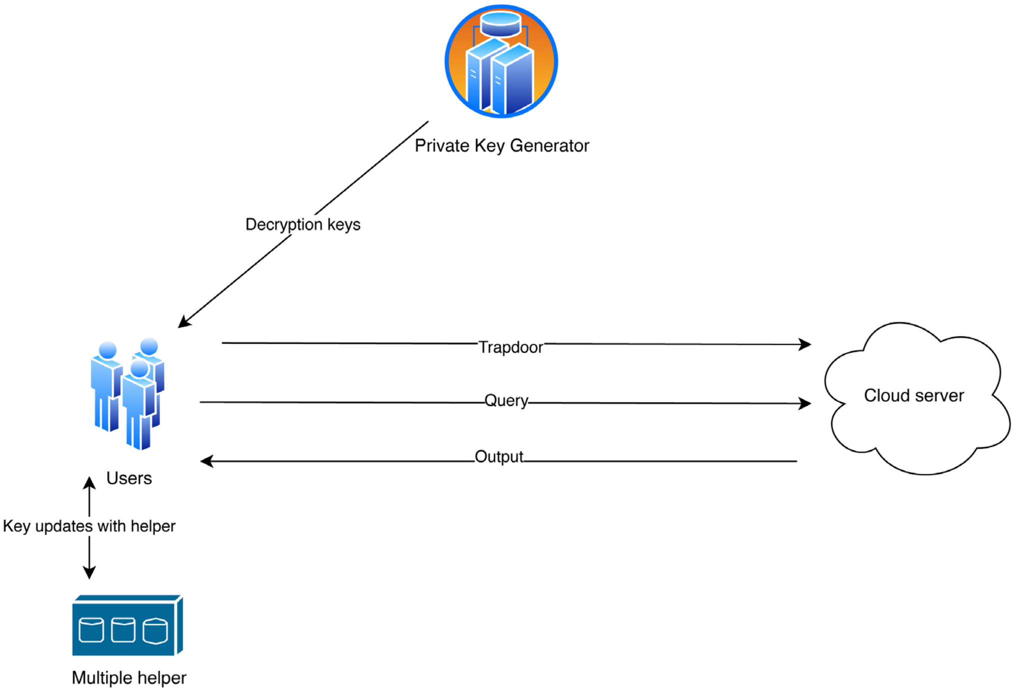

Figure 1. System model of PKI-IBPKE-ET.

1) **Setup** (λ, N, Q): It input a secured parameter λ, total time period *N*, numbr of helper keys Q and returns the public parameter *K*, helper keys $(bk_0, \cdots, bk_{Q-1})$ and the temporal master key *MSK*.

2) **Extract** (MSK, K, ID): On input *MSK*, arbitrary $ID \in \{0,1\}^*$, system parameter K and returns a secret key mdk_{ID_0} to the user with a corresponding identity *ID*. The PKG also performs such algorithm. Subsequently, PKG send to the user with a corresponding identity *ID* via a dedicated secure channel.

3) **UserKeyGeneration** (K, N, mdk_{ID}): The user generation algorithm on input the received secret key mdk_{ID}, public paramter *K*, time period *N* and *ID*. The algorithm output helper key BK_0.

4) **BaseKeyUpdate** (BK_0, bk_j, t): On input the helper key BK_0 at a span bk_j and index time span *t*. The algorithm output update key UK_t.

5) **UserKeyUpdate** $(mdk_{ID_{t-1}}, t, UK_t)$: On input $mdk_{ID_{t-1}}$, index *t* of the next span and update key UK_t. It output the secret key mdk_{ID_t} for the next span t corresponding to the user *ID*.

6) **PKIEncrypt** (K, t, ID, M_1): It input *K*, the index span *t* of the current time period, an identity $ID \in \{0,1\}^*$ and plaintext $M_1 \in M$, and return the ciphertext CT_t as $CT_t = (t, CT_1)$, where $CT_t \in CT$.

7) **PKIDecryption** (mdk_{ID_t}, t, CT_t): It takes a current private secret key mdk_{ID_t} and ciphertext CT_t as input and return plaintext $M_1 \in M$ or a symbol $\perp$ if the corresponding ciphertext is valid.

8) **Test** (CT_{t_A}, CT_{t_B}): It takes ciphertext CT_{t_A} and CT_{t_B} outputted by user *A* and user *B* respectively. It output 1 if the corresponding message coorresponding to CT_{t_A} and CT_{t_B} are equal. It output 0, otherwise $\perp$.

Correctness:

1) When mdk_{ID_t} the updated secret decryption key is generated with multiple helper. The BaseKeyUpdate algorithm on input *ID* as the public key, then;

$$\forall M_1 \in M : Decrypt(CT, mdk_{ID_t}) = M_1,$$

where $CT = Encrypt(ID, M_1)$ and $CT_t = (t, CT)$.

2) Supposedly, tdr_A and tdr_B are trapdoors generated by the trapdoor algorithm given ID_A and ID_B as the public keys, then;

$$\forall M_1 \in M : Test(CT_A, td_A, CT_B, td_B) = 1,$$

where $CT_A = Encrypt(ID_A, M_1)$ and $CT_B = Encrypt(ID_B, M_1)$.

3) The tdr_A and tdr_B are supposedly trapdoors generated by the trapdoor algorithm given ID_A and ID_B as the public keys, then:

$$\forall M_1, M_1' \in M \quad \text{and} \quad M_1 \neq M_1^*. \ \ Pr\left[Test(CT_A, td_A, CT_B, td_B) = 1\right]$$

is negligible where $CT_A = Encrypt(ID_A, M_1)$ and $CT_B = Encrypt(ID_B, M_1')$

4. Security Models

1) **Setup** (λ): The challenger on input a security parameter λ executes the setup algorithm. It gives the system parameters *K* to the adversary *A* and keep

the master key MSK to himself.

2) **Phase 1**-Private secret decryption key queries (ID_A): The challenger runs the extract algorithm to generate the private decryption key mdk_{t_a} corresponding to the user with public key ID_a. It forwards mdk_{t_a} to A.

3) **Trapdoor Queries** (ID_a): The challenger executes the above private decryption key queries on ID_a to obtain mdk_{ID_a} and subsequently generate the trapdoor tdr_a using mdk_{ID_a} via trapdoor algorithm. Finally, the algorithm forwards td_a to A.

4) **Decryption Queries** $(ID_a,(t,CT_a))$: The challenger executes decryption algorithm to decrypt the ciphertext (t,CT_a) by executing extract algorithm to obtain the private secret key $mdk_{ID_{t_a}}$ relating to the public key ID_a. Finally, it forwards plaintext M_1 to A.

5) **Challenge**: A submits an identity ID_{ch} to which a challenge will be posed. The only constraints is that ID_{ch} was not seen in private decryption key queries in phase 1 but ID_{ch} may show in trapdoor queries in phase 1 or in decryption query ID_{ch}. The challenger then randomly chooses plaintext $M_{ch} \in M$ and sets $CT^* = Encrypt(ID_{ch}, M_{ch}, tok^*_{ID})$. Finally, it forwards CT^* to A as its challenge ciphertext.

6) **Phase 2**-Private decryption key queries ID_a: Whereby $ID_a \neq ID_{ch}$. The challenger respond similar to that of phase 1.

7) **Trapdoor Queries** $(ID_a,CT_i) \neq (ID_{ch},CT^*)$: The challenger then responds similar to phase 1.

8) **Decryption Queries** $(ID_a,CT_i) \neq (ID_{ch},CT^*)$: The challenger respond similar to phase 1.

9) **Guess**: A submits a guess $M_1' \in M$

The scheme is W-ID-CCA secure if for all W-IND-CCA adversaries,

$$ADV^{W\text{-}ID\text{-}CCA}_{PKI\text{-}IBPKE\text{-}ET_A}(K) = Pr\left[M_1 = M_1^*\right] \tag{2}$$

is negligible.

5. Construction

The detailed construction for the PKI-IBPKE-ET in this section includes:

1) **Setup**: (λ,Q,N) The system input a secured parameter λ, number of helper key Q, a time period N as input and return public system parameter K. The initial master secret key is MSK and multiple helper keys are $(bk_0,\cdots,bk_{Q-1})$.

- The system generates two multiplicative groups G and G_T with the same prime order p of λ length bits and a bilinear map $e: G \times G \to G_T$. The system selects an arbitrary generator $g \in G$.
- The algorithm exploits a keyed permutation $F: \{0,1\}^k \times \{0,1\}^n \to Z_p^*$ for a positive integers $K = k(\lambda)$ and $L = (n(\lambda))$. Set a random value k_1 from $\{0,1\}^L$. Generate a MAC scheme $MAC = GSV$, where G is generate, S is sign and V verify. It obtain k_2 by running $G(\lambda)$. Set the master token key $MTK = (k_1,k_2)$.
- The system chooses three hash functions: $H_1: \{0,1\}^p \to Z_p^*$, $H_2: \{0,1\}^* \to G$,

$H_3 : T \times G_T \to \{0,1\}^{p+l}$ where l is the length of random numbers, whereas p is the message length. The algorithm randomly picks $(\alpha,\beta) \in Z_p^2$ and set $g_1 = g^{\alpha}$, $g_2 = g^{\beta}$. It publishes public parameter $K = (T, p, G, G_T, e, g, g_1, g_2, bk_Q, MAC, H_1, H_2, H_3)$ and $MSK = (\alpha,\beta)$. T is referred to as MAC tag.

2) **Extract** (K, MSK, ID): For a given string $ID \in \{0,1\}^*$, public parameter K and *MSK*. The algorithm compute $h_{ID} = H_2(ID) \in G$, set temporal master decryption key $mdk_{ID_t} = \left(h_{ID_t}^{\alpha}, h_{ID_t}^{\beta}\right)$ where (α,β) are the master secret key and the intial time index period at *t*.

3) **UserKeyGeneration** (K, mdk_{ID_t}, ID_t): On input mdk_{ID_t}, the algorithm randomly chooses $bk_{Q-1} \in \{0,1\}^p$ and set:

$$BK_0 = g^{bk_{Q-1}}, \quad g_3 = g^{\alpha}\left(\prod_{i=1-Q}^{0}\left(g^{H_1(bk_j(i))}\right)^{r_1}\right), \quad g_4 = \left(\prod_{i=1-Q}^{0}\right), \tag{3}$$

where $r_1 = F(bk_j, i)$ and $j = (1 \bmod Q)$.

The function F is assumed as a pseudorandom permutation.

The initial secret helper keys $BK_0 = (g_3, g_4)$ and number of helper set to $(bk_0, \cdots, bk_{Q-1})$

4) **BaseKeyUpdate** (bk_j, t): On input helper key at bk_j and a period index *t*. The helper key updater computes the *j*th helper base as:

$$UK_t = \left(g_3^{H_1(bk_j(t-Q))}, g^{r_t - Q}\right), \tag{4}$$

where $r_t = F(bk_j, t-Q)$ and $j = (t \bmod Q)$.

5) **UserKeyUpdate** (t, UK_t, mdk_0, ID): On input the period *t*, updated key at time *t* and a master decryption key with $ID \in \{0,1\}^*$. The algorithm parse:

$$UK_t = \left(H_{1_t}, H'_{1_t}\right) \text{ and set } UTKU_{t-1} = \left(g_{3_t}, g_{4_t}\right),$$

$$g_{4_{t-1}} = g_{4_{t-1}} \cdot H_{1_t}, \quad g_{3_{t-1}} = g_{3_{t-1}} \cdot H'_{1_t}.$$

Hence $UTKU_t = \left(g_{ID_{4_t}}, g_{ID_{3_t}}\right)$. Thus, $g_3 = g^{\alpha}\left(\prod_{i=1-Q}^{t}\left(g^{H_1(bk_j(i))}\right)^{r_1}\right)$ and

$j = (i \bmod Q)$, $g_4 = \left(\left(\prod_{i=1-Q}^{0} g^{r_1}\right)^{\beta}\right)$, where $r_1 = F(bk_j, i)$.

The algorithm parse the current index period secret decryption key as:

$$mdk_{ID_t} = \left(h_{ID_t}^{\alpha}, h_{ID_t}^{\beta}\right), \tag{5}$$

where $g_3 = h_{ID}^{\alpha(t)}$ and $g_4 = h_{ID}^{\beta(t)}$.

6) **PKITrapdoor** (ID, MSK, t): For a given string $ID \in \{0,1\}^*$, *MSK* and index time t the algorithm computes $h_{ID} = H_2(ID) \in G$ and set the trapdoor $td_{ID} = h_{ID}^{\beta}$, td_{ID} is the second element of mdk, where mdk_{ID_t}, td_{ID} and tok_{ID} are distributed via a secured channel.

7) **PKIEncrypt** (K, ID, M_1): To encrypt M_1 with a public identity *ID*, the algorithm selects two random numbers $(r_1, r_2) \in Z_p^*$. Then it computes:

$$CT_1 = g^{r_1}, \quad CT_2 = Q_1^{r_1} \cdot H_2\left(e\left(g_4, h_{ID}\right)^{r_1}\right)$$

where

$$Q_1 = \left(\left(\prod_{i=1-Q}^{t} BK_j^{H_1(i)}\right) \cdot M_1\right), \quad CT_3 = g^{r_2},$$

$$CT_4 = \left(M_1 \parallel r_1\right) \oplus H_3\left(CT_1 \parallel CT_2 \parallel P \parallel e\left(g_3, h_{ID}\right)^{r_2}\right).$$

Finally, it returns

$$CT = \left(CT_1, CT_2, CT_3, CT_4\right).$$

where $P \leftarrow S\left(k_2, CT_3\right)$ for signing algorithm S of the employed MAC, the corresponding tag P is used to verify CT_3. The function F is assumed to be a strong pseudorandom permutation and MAC is existentially unforgeable under chosen message attack.

8) **PKIDecrypt** $\left(CT, mdk_{ID}, tok_{ID}\right)$: On input the ciphertext CT, updated secret key mdk_{ID} and a token $token = \left(k_1, k_2\right)$ subsequantly, it computes:

$$m' \parallel r' = CT_4 \oplus H_3\left(CT_1 \parallel CT_2 \parallel P \parallel e\left(CT_3, mdk_{ID_t}^{\alpha}\right)\right),$$

$$m' \parallel r' = H_3\left(e\left(CT_3, mdk_{ID_t}^{\alpha}\right)\right).$$

Given $P \leftarrow S\left(k_2, CT_3\right)$ where $P = MAC_{k_2}\left(CT_3\right)$, the algorithm veriifies if: $B' = MAC_{k_2}\left(CT_3\right)$ if $B' = P$. Then it checks whether $CT_1 = g^{r_1'}$ and $CT_2 = Q_1^{r_1'} \cdot H_2\left(e\left(CT_1, h_{ID_t}^{\beta}\right)\right)$. Where $Q_1 = \left(\prod_{i=1-Q}^{t} BK_j^{H_1(i)}\right) \cdot M_1'$.

If both hold, the algorithm returns M_1', otherwise return $\perp$

9) **Test** $\left(CT_A, td_{ID_A}, CT_B, td_{ID_B}\right)$: On input the ciphertext CT_A, trapdoor td_A and a given senders ciphertext CT_B. The algorithm test whether $M_{1_A} = M_{1_B}$ by computing:

$$T_A = \frac{CT_{2_A}}{H_2\left(e\left(CT_{1_A}, td_{ID_A}\right)\right)}, \quad T_B = \frac{CT_{2_B}}{H_2\left(e\left(CT_{1_B}, td_{ID_B}\right)\right)}. \tag{6}$$

The algorithm output 1 if the above corresponding equation holds, it output 0 otherwise.

Correctness:

The requirement for the above definition is shown below:

1) The first point is verifiable and straightforward as shown above.

2) With a well-formed ciphertext for ID_A and ID_B. Given the following:

$$T_A = \frac{CT_{2_A}}{H_2\left(CT_{1_A}, td_{ID_A}\right)}, \quad T_B = \frac{CT_{2_B}}{H_2\left(CT_{1_B}, td_{ID_B}\right)}$$

$$T_A = \frac{Q_{1_A}^{r_{1A}} \cdot H_2\left(e\left(g_A^{r_1}, h_{ID_A}^{\beta(t)}\right)\right)}{H_2\left(e\left(g_A^{r_1}, h_{ID_A}^{\beta(t)}\right)\right)}, \quad T_B = \frac{Q_{1_B}^{r_{1B}} \cdot H_2\left(e\left(g_B^{r_1}, h_{ID_B}^{\beta(t)}\right)\right)}{H_2\left(e\left(g_B^{r_1}, h_{ID_B}^{\beta(t)}\right)\right)}$$

$$T_A = Q_{1_A}^{r_{1A}} \text{ and } T_B = Q_{1_B}^{r_{1B}}.$$

The algorithm output 1 if the following corresponding equation holds. Otherwise, it output 0.

$$e\left(CT_{1_A}, T_B\right) = e\left(CT_{1_B}, T_A\right).$$

Therefore:

$$\left(CT_{1_A}, T_B\right) = e\left(g^{r_{1A}}, Q_{1_B}^{r_{1B}}\right) = e\left(g, Q_{1_B}\right)^{r_{1A} r_{1B}}$$

$$e\left(CT_{1_B}, T_A\right) = e\left(g^{r_{1B}}, Q_{1_A}^{r_{1A}}\right) = e\left(g, Q_{1_A}\right)^{r_{1A} r_{1B}}.$$

where

$$Q_{1_A} = \left(\left(\prod_{i=1-Q}^{t} BK_j^{H_1(i)}\right) \cdot M_{1_A}\right) \text{ and } Q_{1_B} = \left(\left(\prod_{i=1-Q}^{t} BK_j^{H_1(i)}\right) \cdot M_{1_B}\right).$$

Given the token $tok_{ID} = k_1$, the function output M_A and M_B

If $Q_{1_A} = Q_{1_B}$, then: $e\left(CT_{1_A}, T_B\right) = e\left(CT_{1_B}, T_A\right)$.

Test $\left(CT_A, td_{ID_A}, CT_B, td_{ID_B}\right)$ output 1.

3) For any $M_A \neq M_B$, **Test** $\left(CT_A, td_{ID_A}, CT_B, td_{ID_B}\right) = 1$. This implies that:

$$e\left(g, Q_{1_A}\right)^{r_{1A}} = e\left(g, Q_{1_B}\right)^{r_{1B}}.$$

Hence,

$$Pr\left[e\left(g, Q_{1_A}\right) = \left(g, Q_{1_B}\right)\right] = \frac{1}{2}.$$

Therefore, we assume:

$$Pr\left[Test\left(CT_A, td_{ID_A}, CT_B, td_{ID_B}\right) = 1\right]$$

is negligible.

6. Security Analysis

The PKI-IBPKE-ET scheme is W-IND-ID-CCA secure using the random oracle model assuming Bilinear Diffie-Helman Problem (BDHP) is negligible.

Proof Theory: It is assumed $\mathbb{A}$ is a probabilistic polynomial time (PPT) adversary attacking the W-IND-CCA security of our scheme. Supposedly, $\mathbb{A}$ executes in time T and issues hash queries (q_H) and decryption queries (q_H). Let $Adv_{A}^{W\text{-}IND\text{-}CCA}\left(t, q_H, q_D\right)$ depicts the benefit of $\mathbb{A}$ in W-IND-ID-CCA experiment.

Our proof of security is similar to [3]. The preliminaries of the original game are outlined as follows:

1) **Game G_0**

- $\alpha \leftarrow Z_p^*$, $g_1 = g^{\alpha}$, $T = N$, $BK = \left\{bk_0, \cdots, bk_{Q-1}\right\}$, $R = \varnothing$.
- $M_1 \leftarrow G$, $r_0 \leftarrow Z_p^*$, $U_0^* = g^r$, $V_0^* = M_1^r$,
  $W_0^* = H\left(T, \left(bk_{Q-1}\right)^*, U_0^*, V_0^*, g_1^r\right) \oplus \left(M_1 \| r\right)$.
- $M_1 \leftarrow \mathbb{A}^{o_H, o_2}\left(T, \left(bk_{Q-1}\right)^*, U_0^*, V_0^*, W_0^*\right)$, where the oracle works as follows:
- O_H: On the tuple: $\left(T, \left(bk_{Q-1}\right), U_0, V_0, Y_0\right) \in G^4$, where a same random value

is returned, the same input could be asked multiple times but the same answer will be responded to.

- O_2 : On input a ciphertext $\left(T,\left(bk_{Q-1}\right),U_0,V_0,W_0\right)$, it returns the decryption algorithm to decrypt it using the secret key α given within an index time N and a helper key *Q*.

Let X_o be the event that $M_1' = M_1$ in Game G_0. However the probability in Game G_0 is $Pr\left[S_o\right]$. Hence, we modify Game G_0 and obtain the proceeding game.

2) **Game** $\boldsymbol{G_1}$

- $\alpha \leftarrow Z_p^*$, $g_1 = g^\alpha$, $T = N$, $BK = \left\{bk_0,\cdots,bk_{Q-1}\right\}$, $R = \varnothing$.
- $M_1 \leftarrow G$, $r_0 \leftarrow Z_p^*$, $U_0^* = g^r$, $V_0^* = M_1^r$, $R_0^* \rightarrow [0,1]^{p+i}$,

  $W_0^* = H\left(T,\left(bk_{Q-1}\right)^*,U_0^*,V_0^*,g_1^r\right) \oplus \left(M_1 \parallel r\right)$,

  $R_0 = R_0 \cup \left(T,\left(bk_{Q-1}\right)^*,U_0^*,V_0^*\left(U_0^*\right)^\alpha,R_0^*\right)$.
- $M_1 \leftarrow A^{O_H,O_2}\left(g_1,\left(bk_{Q-1}\right)^*,T,U_0^*,V_0^*,W_0^*\right)$, where the oracle works as:
- O_H : On input a triple $\left(T,\left(bk_{Q-1}\right),U_0,V_0,Y_0\right) \in G^4$ where if there is an entry $\left(T,\left(bk_{Q-1}\right),U_0,V_0,Y_0,h\right)$ in the hash table *R*, *h* is returned, otherwise a random value *h* is selected and returned.

  $$\left(T,\left(bk_{Q-1}\right),U_0,V_0,Y_0,h\right) \text{ is added to } R.$$
- O_2 : On input a ciphertext $\left(T,\left(bk_{Q-1}\right),U_0,V_0,W_0\right)$, a hash query on $\left(T,bk_{Q-1},U_0,V_0,U_0^\alpha\right)$ is issued. Assuming the answer is $h \in [0,1]^{p+i}$, then $M_1 \parallel r$ is computed as $h \oplus W$, then a validity check on whether $U_0 = g^r$ and $V_0 = M_1^r$ is executed. If it fails, $\perp$ is returned: otherwise, M_1 is returned.

The event that **Game** G_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:

$$Pr\left[S_1\right] = Pr\left[S_0\right].$$

We subsequently modify the next game simulation in an indistinguishable way:

3) **Game** G_2

- $\alpha \leftarrow Z_p^*$, $g_1 = g^\alpha$, $T = N$, $BK = \left\{bk_0,\cdots,bk_{Q-1}\right\}$, $R = \varnothing$.
- $M_1 \leftarrow G$, $r_0 \leftarrow Z_p^*$, $U_0^* = g^r$, $V_0^* = M_1^r$, $W_0^* \rightarrow [0,1]^{p+i}$,

  $R^* \rightarrow [0,1]^{p+i}$, $W_0^* = H\left(T,\left(bk_{Q-1}\right)^*,U_0^*,V_0^*,g_1^r\right) \oplus \left(M_1 \parallel r\right)$,

  $R_0 = R_0 \cup \left(t,\left(bk_{Q-1}\right)^*,U_0^*,V_0^*\left(U_0^*\right)^\alpha,W_0^*\right)$.
- $M_1 \leftarrow A^{O_H,O_2}\left(g_1,T,\left(bk_{Q-1}\right),U_0^*,V_0^*,W_0^*\right)$.

The oracle response to queries as follows:

- O_H : Game G_2 is identical to Game G_2. However if adversary queries for $\left(U_0^*,.,\left(U_0^*\right)^\alpha\right)$, then the game is abrogated. $\mathcal{E}$ represents this event.
- This is also the same as Game G_1, however if adversary ask for decryption of $\left(U_0^*,V_0^*W_0\right)$, where $W_0' \neq W_0^*$, $\perp$ is returned.

Chosen Ciphertext security (CCA) secure is paramount in this game because

W_0^* is a random value in both Games, however the random oracle responds are unique and probabilistic because W_0^* is dependent on U_0 and V_0^*. The probability of $\perp$ occurring is negligible.

We modify the simulation game in index time period with multiple helper or base key indistinguishable way in the proceeding game.

4) **Game G_3**

- $\alpha \leftarrow Z_p^*$, $g_1 = g^\alpha$, $T = N$, $BK = \{bk_0, \cdots, bk_{Q-1}\}$, $R = \varnothing$, $t \in N$.
- $M_1 \leftarrow G$, $r_0 \leftarrow Z_p^*$, $U_0^* = g^r$, $V_0^* = M_1^r$, $W_0^* \to [0,1]^{p+i}$,
  $R_0 = R_0 \cup \left(t, \left(bk_{Q-1}\right)^*, U_0^*, V_0^* \left(U_0^*\right)^\alpha, W_0^*\right)$.
- $M_1 \leftarrow \mathbb{A}^{O_H, O_2}\left(g_1, T, \left(bk_j\right), U_0^*, V_0^*, W_0^*\right)$.
- O_H: Game G_3 is identical to Game G_2. However if adversary queries for $\left(U_0^*, T, bk_j, U_0^*, ., \left(U_0^*\right)^\alpha\right)$, then the game is abrogated. Let ε_1 be this event.
- This is also the same as Game G_2, however if adversary ask for decryption of $\left(U_0^*, bk_j, V_0^*, t\right)$ where $bk_j' \neq bk_j$, $\perp$ is returned.

The timestamp and the base key $\left(bk_j\right)$ at a period *j* associated with the ciphertext improve the security of this game. *t* is a timestamp value associated with the ciphertext in both Games, however the random oracle response are unique and probabilistic because decryption queries are dependent on T, U_0^*, V_0^* and $\left(bk_j\right)$. The probability of $\perp$ occurring is negligible.

In this game, the challenge ciphertext identically distributed in Game G_2 and G_3 as W_0^* is a chosen random value in both Game G_2 and Game G_3. The simulation of O_2 is secure since W_0^* is uniquely determined by U_0^* and V_0^* in Game G_2 and U_0^*, V_0^*, T and bk_j in Game G_3. Therefore, if event ε_1 does not occur, Game G_3 is identical to Game G_1. However, it is observed 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**: $\left(T, U_0, V_0, U_0^\alpha\right)$ is queried to O_H before a decryption query $\left(T, U_0, V_0, W_0\right)$ is issued.

In this case, W_0 is determined after $\left(T, U_0, V_0, U_0^\alpha\right)$ is queried to O_H. So the decryption oracle is perfectly simulated.

- **Type 2**: $\left(U_0, V_0, U_0^\alpha\right)$ is not queried to O_H when a decryption query $\left(U_0, V_0, W_0, BK\right)$ was issued. Subsequently, $\perp$ is returned by the decryption oracle. The simulation will fail if $\left(U_0, V_0, W_0, BK\right)$ is valid. Therefore, this happens with negligible probability.

7. Comparison

The efficiency of algorithms and time consumption of our scheme is compared with: Ma's [8] scheme, which combined the public key cryptosystem with equality test and identity-based cryptographic primitive: Wu *et al.*'s [29] scheme, which proposed a scheme to resist the insider attack: and Li *et al.* [30] scheme. Our scheme unveils a parallel key-insulation cryptosystem with multiple helper to minimize the exposure of the helper keys and decryption keys. The comparative results on the efficiency of our method is shown in **Table 1** and communication cost in **Table 2**. The above comparison shows that our scheme can resist IA with key-insulation, whereas others' don't have this ability. In addition, the schemes [8] [29] as well as our scheme implement chosen ciphertext security, which is stronger than chosen plaintext security [16] [30] and other related identity based parallel key-insulated primitive [16] via the random oracle model. To evaluate computation efficiency of these schemes, pairing-based cryptography (PBC) library [31] was used to quantify the time consumption of encryption, decryption and test operations. We examine the computational efficiency in these schemes, the Pairing-Based Cryptography (PBC) Library [31] is used to quantify the time consumption of encryption, decryption and test operations. We use the code of a program in VC++ 6.0 and executed on a computer (Windows 10 Pro, operating system), Capacity of Intel(R) Core (TM) i5-4460 CPU with 3.20 GHZ and 4Gb RAM. The code was executed several times and average time of execution extracted. With respect to the scheme and other pairing-based constructions with a security level of 1024-bit RSA, a supersingular curve $z^2 = x^3 + x$ with an embedded degree of 2 is adopted. Also, $q = 2^{159} + 2^{17} + 1$ noted as a 160-bit Solinas prime with $p = 12qr - 1$ noted as a 512-bit prime. With regards to ECC-based schemes, an equivalent security level of Koblitz elliptic curve of $y = x^3 + ax^2 + b$ defined on a $F_{2^{163}}$ is used to provide the same security level in the ECC group. The computational units are in millisecond (ms)

Table 1. Efficiency comparison of algorithm of variant PKE-ETs.

SCH	PKI	IA	Encryption	Decryption	Test	Security	R	TD	ET
[8]	N	N	$4Exp_1 + 2Exp_2$	$2P + 2Exp_1$	4P	OW-ID-C	Y	Y	Y
[29]	N	Y	$1P + 4Exp_1 + 2Exp_2$	$1P + 2Exp_1$	2P	W-I-ID-C	Y	N	Y
[30]	N	N	$1P + 4Exp_1 + 1Exp_2$	3P	4P	I-ID-C	Y	N	N
Ours	Y	Y	$2P + 2Exp_1 + 2Exp_2$	$2P + 2Exp_1$	2P	W-I-ID-C	Y	Y	Y

Legends: In this table, "SCH": scheme, "Exp_1" and "Exp_2": exponent computation in group 1 and group 2, "P": pairing computation, "PKI": parallel key-insulated, "IA": insider attack, "R": random oracle model, "TD": trapdoor, "ET": equality test, "Y": "Yes" as a supportive remark, "N" refers to "No" as not supportive, "I": IND, "C_A": CPA, "C": CCA.

Table 2. Communication cost comparison.

SCHEME	PK_{size}	SK_{size}	CT_{size}	Del_{Auth}	ROM	Assumption
[8]	$2G_0$	$2Z_{p_0}$	$4G_0 + Z_{p_0}$	Yes	Yes	BDH
[29]	$2G_0$	$3Z_{p_0}$	$4G_0 + Z_{p_0}$	No	Yes	BDH
[30]	$2G_0$	$2Z_{p_0}$	$2G_0 + Z_{p_0}$	No	Yes	BDH
Ours	$2G_{p_0}$	$2Z_{p_0}$	$2G_0 + Z_{p_0}$	Yes	Yes	BDH

Legends: In this table, PK_{size}; size of public key, SK_{size}; size of secret key, CT_{size}; size of ciphertext, Del_{Auth}: authorization, BDH; bilinear Diffie-Hellman, G_0: group G, Z_{p_0}; Z_p, ROM: random oracle model. W-IND-ID-CCA refers to weak indistinguishable chosen ciphertext attack against identity, OW-ID-CCA refers to one-way chosen ciphertext attack against identity and IND-ID-CPA refers to indistinguishable chosen plaintext attack against identity.

and bytes respectively. The execution times of each respective algorithm were calculated and Matlab program was used to generate **Figure 2**. The Figure (see **Figure 2**) depicts the computation cost of decryption and test of our scheme comparable with other existing works, whereas our encryption computational cost seems higher. This is reasonable due to the additional computational overheads required to prevent helper keys exposure with the adoption of multiple helpers, 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 [29] [30]. Although time consumption of encryption and decryption operations of our scheme is higher than scheme proposed in [29], our scheme provides additional security to helper exposure.

Furthermore, our computational overhead cost results do not make our scheme superior to other related schemes in terms of computational cost analysis. However, this is pardonable due to the fact that additional computational cost values added to our scheme increases the computational variables. In this way, the computational cost in encryption and decryption are higher than the related scheme due to the extra multiple helper computation added to our scheme. However, the test computational cost is comparable to [8], but the other related scheme has a high computational test results. Therefore, our scheme is an improvement on the related public key cryptosystem with key-insulation. However, our scheme adds additional primitives on previous schemes such as equality test and the adoption of multiple helper to improve on the use of single or double helper in protecting decryption keys. In this way, our scheme is secure against the use of single helper in updating users decryption keys in key-insulated cryptosystems. Therefore, the superiority of our scheme is achieved in the lower pairing computations, insider attack resistant with delegated equality test, and a symmetric cryptographic primitive (MAC) addition to public key cryptosystem to construct our scheme.

8. Conclusion

This paper introduced a scheme to solve the problem caused by private decryption

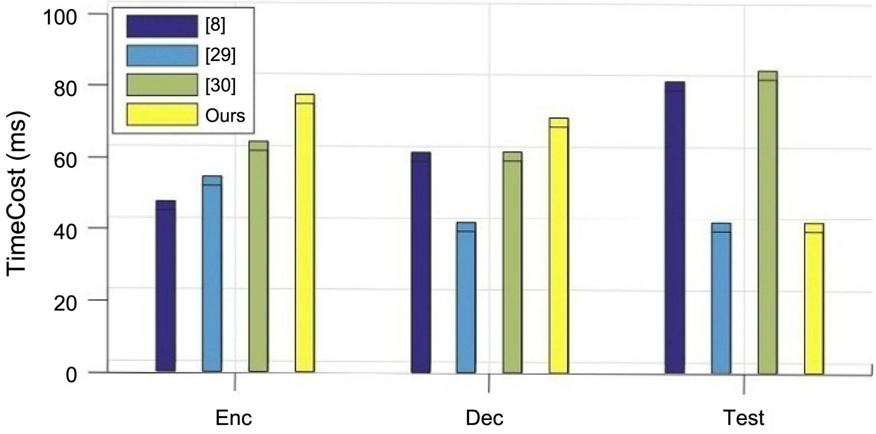

Figure 2. Computational overhead of related schemes.

key exposure and helper key in identity based cryptosystem with equality test. Our scheme delegates equality test to the cloud server and also thwarts the insider attack phenomenom in public key encryption. Inspired by the notion of scheme in [8] and the use of a single helper with key-insulation [32], we put forward parallel key insulated ID-based public key cryptographic primitive with outsourced equality test (PKI-IBPKE-ET). The mechanism of parallel key-insulated with multiple helper was used to reduce the damage to helper keys and private key exposure. Besides, our scheme also has the ability to resist insider attack from semi-trusted cloud server, which makes it practical and suitable in cloud computing. Our scheme is proven secured in random oracle model. Theoretical analysis and experiment simulation both demonstrate that our scheme is secure and efficient.

Acknowledgements

Sincere thanks to the anonymous reviewers for their kind consideration and a special thanks to managing editor *Hellen XU* for a rare attitude of high quality.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Cao, X., Li, H., Dang, L. and Lin, Y. (2017) A Two-Party Privacy Preserving Set Intersection Protocol against Malicious Users in Cloud Computing. *Computer Standards and Interfaces*, **54**, 41-45. https://doi.org/10.1016/j.csi.2016.08.004

[2] Boneh, D. and Franklin, M. (2001) Identity-Based Encryption from the Weil Pairing. In: *Annual International Cryptology Conference*, Springer, Berlin, 213-229. https://doi.org/10.1007/3-540-44647-8_13

[3] Yang, G., Tan, C.H., Huang, Q. and Wong, D.S. (2010) Probabilistic Public Key Encryption with Equality Test. In: *Cryptographers' Track at the RSA Conference*, Springer, Berlin, 119-131. https://doi.org/10.1007/978-3-642-11925-5_9

[4] Lipmaa, H. (2003) Verifiable Homomorphic Oblivious Transfer and Private Equality Test. In: *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, Berlin, 416-433. https://doi.org/10.1007/978-3-540-40061-5_27

[5] Tang, Q. (2012) Public Key Encryption Schemes Supporting Equality Test with Authorisation of Different Granularity. *International Journal of Applied Cryptography*, **2**, 304-321. https://doi.org/10.1504/IJACT.2012.048079

[6] Wu, L., Zhang, Y., Choo, K.K.R. and He, D. (2017) Efficient Identity-Based Encryption Scheme with Equality Test in Smart City. *IEEE Transactions on Sustainable Computing*, **3**, 44-55. https://doi.org/10.1109/TSUSC.2017.2734110

[7] Lee, H.T., Wang, H. and Zhang, K. (2018) Security Analysis and Modification of ID-Based Encryption with Equality Test from ACISP 2017. In: *Australasian Conference on Information Security and Privacy*, Springer, Cham, 780-786. https://doi.org/10.1007/978-3-319-93638-3_46

[8] Ma, S. (2016) Identity-Based Encryption with Outsourced Equality Test in Cloud

Computing. *Information Sciences*, **328**, 389-402. https://doi.org/10.1016/j.ins.2015.08.053

[9] Dodis, Y., Katz, J., Xu, S. and Yung, M. (2002) Key-Insulated Public Key Cryptosystems. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, 65-82. https://doi.org/10.1007/3-540-46035-7_5

[10] Bellare, M. and Palacio, A. (2006) Protecting against Key-Exposure: Strongly Key-Insulated Encryption with Optimal Threshold. *Applicable Algebra in Engineering, Communication and Computing*, **16**, 379-396. https://doi.org/10.1007/s00200-005-0183-y

[11] Wang, Y., Yan, D., Li, F. and Xiong, H. (2017) A Key-Insulated Proxy Re-Encryption Scheme for Data Sharing in a Cloud Environment. *IJ Network Security*, **19**, 623-630.

[12] He, L., Yuan, C., Xiong, H. and Qin, Z. (2017) An Efficient and Provably Secure Certificateless Key Insulated Encryption with Applications to Mobile Internet. *IJ Network Security*, **19**, 940-949.

[13] Hanaoka, Y., Hanaoka, G., Shikata, J. and Imai, H. (2005) Identity-Based Hierarchical Strongly Key-Insulated Encryption and Its Application. In: *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, Berlin, 495-514. https://doi.org/10.1007/11593447_27

[14] Libert, B., Quisquater, J.J. and Yung, M. (2007) Parallel Key-Insulated Public Key Encryption without Random Oracles. In: *International Workshop on Public Key Cryptography*, Springer, Berlin, 298-314. https://doi.org/10.1007/978-3-540-71677-8_20

[15] Hanaoka, G., Hanaoka, Y. and Imai, H. (2006) Parallel Key-Insulated Public Key Encryption. In: *International Workshop on Public Key Cryptography*, Springer, Berlin, 105-122. https://doi.org/10.1007/11745853_8

[16] Weng, J., Liu, S., Chen, K. and Ma, C. (2006) Identity-Based Parallel Key-Insulated Encryption without Random Oracles: Security Notions and Construction. In: *International Conference on Cryptology in India*, Springer, Berlin, 409-423. https://doi.org/10.1007/11941378_29

[17] Ren, Y. and Gu, D. (2010) CCA2 Secure (Hierarchical) Identity-Based Parallel Key-Insulated Encryption without Random Oracles. *Journal of Systems and Software*, **83**, 153-162. https://doi.org/10.1016/j.jss.2009.07.046

[18] Ren, Y., Wang, S. and Zhang, X. (2013) Practical Parallel Key-Insulated Encryption with Multiple Helper Keys. *Computers and Mathematics with Applications*, **65**, 1403-1412. https://doi.org/10.1016/j.camwa.2012.01.032

[19] Boneh, D., Di Crescenzo, G., Ostrovsky, R. and Persiano, G. (2004) Public Key Encryption with Keyword Search. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, 506-522. https://doi.org/10.1007/978-3-540-24676-3_30

[20] Xu, P., Jin, H., Wu, Q. and Wang, W. (2012) Public-Key Encryption with Fuzzy Keyword Search: A Provably Secure Scheme under Keyword Guessing Attack. *IEEE Transactions on Computers*, **62**, 2266-2277. https://doi.org/10.1109/TC.2012.215

[21] Yu, Y., Ni, J., Yang, H., Mu, Y. and Susilo, W. (2014) Efficient Public Key Encryption with Revocable Keyword Search. *Security and Communication Networks*, **7**, 466-472. https://doi.org/10.1002/sec.790

[22] Qu, H., Yan, Z., Lin, X. J., Zhang, Q. and Sun, L. (2018) Certificateless Public Key Encryption with Equality Test. *Information Sciences*, **462**, 76-92.

https://doi.org/10.1016/j.ins.2018.06.025

[23] Zhang, K., Chen, J., Lee, H.T., Qian, H. and Wang, H. (2019) Efficient Public Key Encryption with Equality Test in the Standard Model. *Theoretical Computer Science*, **755**, 65-80. https://doi.org/10.1016/j.tcs.2018.06.048

[24] Lin, X.J., Sun, L. and Qu, H. (2018) Generic Construction of Public Key Encryption, Identity-Based Encryption and Signcryption with Equality Test. *Information Sciences*, **453**, 111-126. https://doi.org/10.1016/j.ins.2018.04.035

[25] Waters, B. (2005) Efficient Identity-Based Encryption without Random Oracles. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, 114-127. https://doi.org/10.1007/11426639_7

[26] Alornyo, S., Asante, M., Hu, X. and Mireku, K.K. (2018) Encrypted Traffic Analytic Using Identity Based Encryption with Equality Test for Cloud Computing. 2018 *IEEE 7th International Conference on Adaptive Science and Technology*, Ghana, 22-24 August 2018, 1-4. https://doi.org/10.1109/ICASTECH.2018.8507063

[27] Wu, L., Zhang, Y., Choo, K.K.R. and He, D. (2017) Efficient and Secure Identity-Based Encryption Scheme with Equality Test in Cloud Computing. *Future Generation Computer Systems*, **73**, 22-31. https://doi.org/10.1016/j.future.2017.03.007

[28] Lee, H.T., Ling, S., Seo, J.H. and Wang, H. (2016) Semi-Generic Construction of Public Key Encryption and Identity-Based Encryption with Equality Test. *Information Sciences*, **373**, 419-440. https://doi.org/10.1016/j.ins.2016.09.013

[29] Wu, T., Ma, S., Mu, Y. and Zeng, S. (2017) ID-Based Encryption with Equality Test against Insider Attack. In: *Australasian Conference on Information Security and Privacy*, Springer, Cham, 168-183. https://doi.org/10.1007/978-3-319-60055-0_9

[30] Li, J., Zhang, F. and Wang, Y. (2006) A Strong Identity Based Key-Insulated Cryptosystem. In: *International Conference on Embedded and Ubiquitous Computing*, Springer, Berlin, 352-361. https://doi.org/10.1007/11807964_36

[31] Lynn, B. (2013) PBC Library. https://crypto.stanford.edu/pbc/

[32] Alornyo, S., Zhao, Y., Zhu, G. and Xiong, H. (2020) Identity Based Key-Insulated Encryption with Outsourced Equality Test. *IJ Network Security*, **22**, 257-264.