International Journal of Cyber-Security and Digital Forensics (IJCSDF) 7(3): 305-316
The Society of Digital Information and Wireless Communications (SDIWC), 2018 ISSN: 2305-001

Identity-Based Encryption Technique with Ranking for Multi-Keywords Search in Cloud (IDERMKS)

Regina Esi Turkson, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Email: regina_turkson@yahoo.com or rturkson@ucc.edu.gh

Yongjian Liao, School of Information and Software Engineering, University of Electronic Science and Technology of China, Email: liaoyj@uestc.edu.cn

Edward Yeallakuor Baagyere, School of Information and Software Engineering, University of Electronic Science and Technology of China, Email: ybaagyere@uds.edu.gh

Abstract— **Cloud providers use virtualization technologies combined with self-service capabilities for computing resources via network infrastructure. In recent time, massive growth in big data generated through cloud has been noticed. This massive growth of data in the cloud has security issues and challenges. Sensitive data outsourced into the cloud needs to be encrypted and this makes searching and retrieval of the data very difficult since we have voluminous data store in the cloud. In this paper, we address this difficulty by proposing an Identity-based technique with ranking for multi-keywords search (IDERMKS) scheme enabling data users to search for and retrieve encrypted files that has been outsourced into the cloud by a data owner. The retrieved encrypted files are ranked based on their relevance scores and the top-*k* relevant files are then returned by the cloud to the data user. The data user then obtains a decryption key from the appropriate data owner to decrypt the selected file. The security requirement of our scheme is provably secure and the performance of our scheme is also more efficient as compared to other Public-key Encryption with Keyword Search (PEKS) schemes.**

Keywords— Cloud Computing, Identity-Based Encryption, Ranked Multi-Keyword Search, Privacy Preserving

1. INTRODUCTION

As the volume of data keep increasing tremendously, the challenge is not only on how to store and manage these data but also on how to effectively and efficiently analyze these data to gain knowledge in making smart decision [1][2][3] and also on the security and privacy issues of these data.

Cloud computing has recently emerged as the promising technology for handling big data. Cloud computing is a network-based environment that centers on sharing computations or resources. Today, Cloud computing is one of the most thrilling technologies due to its ability to decrease costs relating to computing while increasing flexibility and scalability for computer processes. It is a revolutionary technology that has influenced how computer hardware and software are designed [4]. It provide enormous benefit like on demand and decreased cost of usage, easy access, quick deployment, flexibility, resource management etc. These benefits have accounted for different data owners outsourcing voluminous data onto the cloud. IT organizations have articulated concern about critical security issues that exist with the widespread implementation of cloud computing.

Security in cloud has been one of the most argued-about issues in the field of cloud computing. The risks of compromised security and privacy may be lesser when the data were to be stored on individual machines instead of in cloud. This has made privacy preserving a very important concern in cloud computing. Although cloud service providers are able to deliver highly available storage and massively parallel computing resources, the security and privacy of data stored on the cloud is a big challenge since the server might be curious or might illegally inspect and access user's sensitive data and also unauthorized users may be able to intercept people's data. Outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, however, it does not offer any guarantee on data integrity and availability. This problem may impede the successful deployment of the cloud architecture. Traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted on data as users no longer possess the storage of their data physically. Thus, how to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing. Therefore, to fully guarantee the security of the data and save the cloud users' computation resources, it is of appropriately importance to enable public auditability for cloud data storage so that the users may resort to a Third Party Auditor (TPA), who has the

expertise and capabilities to audit the outsourced data when needed[5].

For the outsourced data not to be leaked to external parties, exploiting data encryption before outsourcing is one way to alleviate this privacy concern. However, without an appropriate designed auditing protocol, encryption itself cannot prevent data from "flowing away" towards external parties during the auditing process. Encryption of the data helps in protecting data confidentiality of the user but searching for data also becomes a challenge. Secure search over encrypted data was first initiated by Song et al [6] .They proposed a cryptographic primitive concept called searchable encryption which enables users to perform a keyword-based search on an encrypted data, just as on plaintext data.

Wang et al [7] were the first to define secure search over encrypted cloud data, however, further development has been made by [8], [9], [10], [11] which incur high storage and computational cost. These researches and more, seeks to reduce computation and storage cost and also enrich the category of search functions such as fuzzy keyword search, secure ranked multi-keyword search and similarity based search but they are limited to single-owner model.

In 1984, Shamir [12] designed a public key encryption scheme in which the public key can be an arbitrary string. Shamir proposed the idea of identity-based cryptography in such a scheme there are four algorithms: Setup, Extract, Encrypt and Decrypt. The notion of identity based cryptosystem is that the public key can be an arbitrary string. Shamir's original motivation for identity-based encryption was to simplify certificate management in e-mail systems. Since the problem was posed in 1984 there have been several proposals for IBE schemes [13][14][15][16]. Boneh and Franklin in 2001, designed the first practical identity-based cryptosystem [17]. Zhang et al [18] defined a multi-owner model for privacy preserving keyword search over encrypted cloud data, however, their scheme did not use an identity based encryption scheme. A similar system was proposed in [19] of which a cryptographic techniques, query, response randomization and ranking capability was used. However, IDE was not used in their system and the approach used in the scheme formulation is quite different from our scheme.

The literatures reviewed above elaborate various concepts on cloud computing, security in cloud, auditable cloud data, searchable encryption, ranked multi-keyword search, privacy preserving, and identity-based cryptography. However, none of these literatures considered combining these concepts.

In our work, we propose an Identity-based technique with ranking for multi-keywords search (IDERMKS) in cloud. The framework of our scheme and its security requirements are defined. We prove that our proposed scheme satisfies the ciphertext and trapdoor indistinguishability in the random oracle [20][21]. Finally, we demonstrate the advantage of our IDERMKS scheme by comparing with previous PEKS schemes.

This paper is organized as follows: Section II outlines the system model, threat model, design goals and the architectural design, Section III defines the preliminaries, Section IV outlines the algorithms and the security requirements of our scheme. We present the proposed IDERMKS scheme and its security analysis in Section V. Section VI, we briefly outline the privacy preserving ranking method which we adopted from [18], Section VII gives the performance analysis of our scheme and we conclude in section VIII.

2. System and Threat Model, Design Goals and Architectural Design

In this section, we describe the system model, threat model, design goals, and gives the architectural design of our proposed scheme.

2.1 System Model

In the proposed IDERMKS scheme, we have three entities namely; data owner, data user and cloud server. The data user has a collection of files that have to be outsourced to the cloud. Before outsourcing, these files need to be encrypted in order to ensure the confidentiality of the files. To enable efficient and adequate search operation on these files, the data owner builds a secure searchable index on keywords sets extracted from the files. The data owner then outsource the encrypted files together with their indexes to the cloud server. When a data user wants to search keywords over these encrypted files, the data user submit the hashed keywords to the data owner which are then used by the data owner to create a trapdoor for the data user. Upon receiving the trapdoor, the data user submit it to the cloud. The cloud server then searches through the encrypted indexes of each data owner and return the corresponding set of files for the indexes that matches and have the same pattern as that of the trapdoor. The cloud then rank these files based on their relevance score and return the top-k important files to the data user. The data user can then obtains a decryption key from the data owner to decrypt the file.

2.2 Threat Model

In the threat model, we assume that the cloud server is honest but curious. The server follows our protocol but keen to know the content of the encrypted files, keywords and the relevance

scores which is the same as in [18]. The data owner and data users who are authenticated by PKI are assumed to be trusted.

2.3 Design Goals

In order to ensure the privacy preserving of the ranked multi-keyword search, the proposed IDERMKS scheme should satisfy some of the security and performance goals in [18]. The proposed IDERMKS should;

a) Enable multi-keyword search over encrypted files from different data owners.
b) Not hinder new data owners from entering the system, thus, providing data user scalability.
c) Ensure that authenticated data users perform correct and appropriate search

2.4 Architectural Design

Our proposed scheme enables a data user to perform a search over encrypted cloud data by making a multi-keyword search query which protect the system-wide privacy in cloud computing. The use of keyword relevance is an intermediate similarity semantic that has been selected among numerous semantics, in which a number of keywords from the search query is used. The appearance of keywords in the file will be used to measure the relevance of that file to the query. The ranking system used in our scheme proves to be very effective and efficient in implementing and returning the highly relevant files to terms submitted in the search query. Ranked based search can appropriately eliminate unnecessary network traffics by returning only the top relevant files which are of user's interest. Figure 1 shows the architecture of the proposed IDERMKS that outlines the various processes and entities of the scheme.

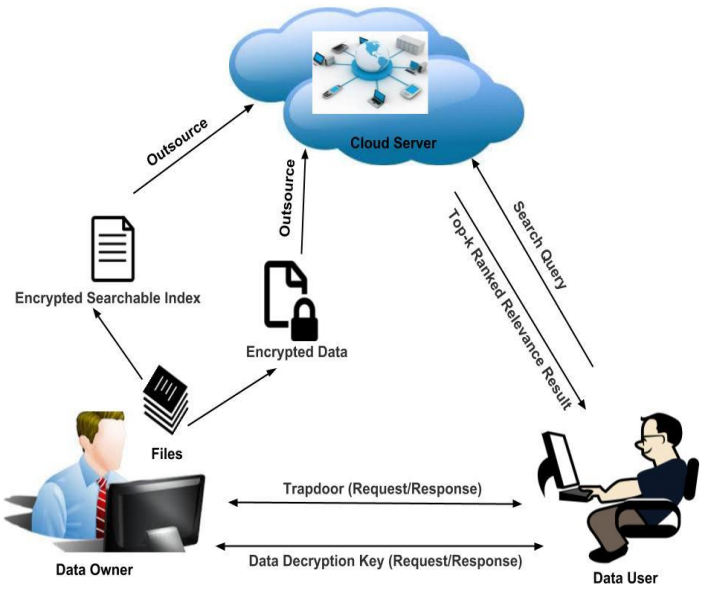

Figure 1: Architecture of IDERMKS

3. PRELIMINARIES

This section gives brief review of the various concepts of bilinear pairing, identity-based encryption and other related mathematical problems used in this paper.

3.1 Bilinear Pairing

Let $(G_1, +)$ and $(G_2, \bullet)$ be cyclic groups of prime order q. Let g_1 and g_2 be the generator of G_1 and G_2 respectively. Let e be a bilinear map defined as $e: G_1 \times G_1 \to G_2$ which satisfies the following conditions:

a) Bilinearity: If $\forall P, Q \in G_1$ and $a, b \in Z_q^*$, then $e(aP, bQ) = e(P, Q)^{ab}$
b) Non-degenerate: $e(P, Q) \neq 1$
c) Computability: There is an efficient algorithm to compute $e(P, Q)$, $\forall P, Q \in G_1$

3.2 Identity-Based Encryption

An identity-based scheme has four algorithms: Setup, Extract, Encrypt and Decrypt.

a) *Setup:* It takes a security parameter k and returns system parameters (*params*) and a master-key. The system parameters include a description of finite message space M and a description of a finite cipher text space C. Ideally, the system parameters will be publicly known, while the master-key will be known only to the Private Key Generator (PKG).
b) *Extract:* It take as input *params*, master-key and an arbitrary string ID which will be used as the public key $ID \in \{1,0\}^*$ and d is the corresponding private decryption key. The extract algorithm extract a private key from the given public key.
c) *Encrypt:* It take $param, ID$ and $\mathrm{M} \in M$ as input and returns $\mathrm{C} \in C$.
d) *Decrypt*: It takes as input $param, \mathrm{C} \in C$ and and private key d and returns $\mathrm{M} \in M$ .These algorithm must satisfy the standard constraint such that if d is the private key generated by the Extract algorithm when given the public key ID ,then $\forall \mathrm{M} \in M$: $\mathrm{Encrypt}(param, ID, \mathrm{M}) = \mathrm{C}$ and $\mathrm{Decrypt}(param, C, d) = \mathrm{M}$

3.3 Other mathematically related problems and assumptions.

Here, we give two mathematical hard problems and also define their corresponding assumptions

a) Bilinear Diffie-Hellman (BDH) problem*:* The BDH problem in $\langle e, G_1, G_2 \rangle$ is defined as follows: Given

$P, aP, bP, cP \in G_1$ for some $a, b, c \in Z_q^*$, compute $e(P,P)^{abc} \in G_2$.

b) Computational Diffie-Hellman (CDH) problem: The CDH problem in $\langle e, G_1, G_2 \rangle$ is defined as follows: Given $P, aP, bP \in G_1$ for some $a, b \in Z_q^*$, the CDH problem is to compute $abP \in G_1$

Definition 1: Given $\langle P, aP, bP, cP \rangle \in G_1$ for some $a, b, c \in Z_q^*$, the BDH assumption is that, there is no probabilistic polynomial-time adversary A with a non-negligible probability that can compute $e(P,P)^{abc} \in G_2$. The adversary A is said to have an advantage $\mathrm{Adv}_{\mathrm{BDH}}(t) = \Pr[A(P, aP, bP, cP) = e(P,P)^{abc}] \geq \varepsilon$ within a running time t.

Definition 2: Given $\langle P, aP, bP \rangle \in G_1$ for some $a, b \in Z_q^*$, the CDH assumption is that, there is no probabilistic polynomial-time adversary A with a non-negligible probability that can compute $abP \in G_1$. The adversary A is said to have an advantage $\mathrm{Adv}_{\mathrm{CDH}}(t) = \Pr[A(P, aP, bP) = abP] \geq \varepsilon$ within a running time t.

4. ALGORITHMS AND SECURITY REQUIREMENTS

We define the algorithms and security requirements for the proposed IDERMKS by modifying the ones in [8] [17] [22].

4.1 Algorithms

An IDERMKS scheme consist of seven (7) algorithms, outlined as follows:

a) *Setup algorithm:* This is a probabilistic algorithm which takes as input, a security parameter l. It then output a master secret key and a master public key $\mathrm{P}_{\mathrm{pub}}$.

b) Key extract algorithm: This is a deterministic algorithm which takes a user's identity, a master secret key and system parameters as input and the output a user Secret key sk_{ID}.

c) IDERMKS algorithm: This is a probabilistic algorithm which takes a public key of the data owner, a set of keywords in a document and system parameters as input and then output IDERMKS ciphertext which is the searchable index I. The data owner encrypts each document with Symmetric-Key Encryption method using different keys for each document. The data owner then encrypts the symmetric keys with a Public Key Encryption which has blinding capabilities and in our case an Identity-Based Encryption method will be used.

d) Trapdoor generation algorithm: This is a probabilistic algorithm which takes as input a multiple keywords, data user's secret key and system parameters and then generate the trapdoor T_w.

e) Test algorithm: This is a deterministic algorithm which takes as input the IDERMKS ciphertext I, trapdoor T_w and system parameters. Upon receiving a query request T_w from the user, the cloud server match the queried keywords from all keywords stored on it against the query request. It then extract all files that contain the query request to obtain a candidate file set.

f) Ranking algorithm: The cloud ranks the files in the candidate file set obtained after running the testing algorithm and find the top-*k* relevant file. An order and privacy preserving encoding scheme which encodes the relevance score to obtain the top-*k* search result will be adopted into our IDERMKS scheme.

g) Retrieval algorithm: This algorithm takes a blinded encrypted symmetric key as input and output a blinded symmetric key which was used to encrypt a document.

4.2 Security Requirements

An IDERMKS scheme must meet the following security requirements: (1) ciphertext indistinguishability and (2) trapdoor indistinquishability.

4.2.1 Ciphertext indistinguishability

We define a security for Indistinguishability of Ciphertext from Ciphertext of Chosen Keyword Attack (IND-CC-CKA) of the IDERMKS scheme. The Game interaction between an adversary *A* and a challenger *C* below defines the security of our scheme.

Game 1 -

a) Setup: The Challenger *C* runs the setup algorithm to generate the master private key and a master public key P_{pub}. *C* generate *A*'s private key sk_{ID_A} by running the key extract algorithm with the corresponding public key for the identity ID_A. The *C*'s master public key P_{pub} and the system parameters are given to the adversary *A*. The challenger then keeps the master private key *msk* and gives *A*'s private key sk_{ID_A} to him.

b) Phase 1: The adversary *A* may make series of different queries to challenger *C* in an adaptive manner as follows:

- *Key generation queries:* The adversary gives an identity ID_A to the Challenger. The challenger returns sk_{IDA} to *A* by running the key extract algorithm.
- *Trapdoor queries: A* makes this queries for keywords W , the challenger *C* returns the trapdoor T_w for keyword W to *A* by running the Trapdoor Generation algorithm.

c) Challenge: The adversary *A* sends $(ID_C^*, W_0^* W_1^*)$ to the Challenger *C* where W_0^* and W_1^* are two challenged keywords. *C* chooses a random value $b \in \{0,1\}$ and uses W_b^* to generate IDERMKS ciphertext I^* by running the IDERMKS algorithm then send the result ciphertext to the adversary *A*.

d) Phase 2: Adversary *A* continues to make the key extract queries for any identity ID_i and the trapdoor query for any keyword W_i to the challenger *C* subject to the restriction that $IDi \neq IDc^*$ and $W_i \neq \{W_0^* \text{ or } W_1^*\}$.

e) Guess: Finally, the adversary *A* output a guess of a value $b' \in \{0,1\}$. *A* wins the game if $b' = b$

From the game above, we define the advantage of the adversary *A* in breaking the ciphertext indistinguishability as the probability that the adversary *A* wins.

Definition 3: An IDERMKS scheme is said to have satisfied ciphertext indistinquishability against adaptive chosen plaintext attack if no adversary have a non-negligible advantage in Game 1.

4.2.2 Trapdoor indistinguishability

This means that an adversary is unable to distinguish the trapdoor for two challenged keywords chosen by him. A Game between a probabilistic polynomial time adversary *A* and a challenger *C* has been defined in Game 2 to represent the trapdoor indistinguishability.

Game 2 –

a) *Setup:* This phase is same as defined in Game 1.

b) *Phase 1:* In this phase, the Adversary A makes a number of dissimilar of queries to the Challenger C in an adaptive manner. This phase is also same as one defined in Game1.

c) *Challenge:* The adversary A sends $(ID_C^*, W_0^* W_1^*)$ to the Challenger C where W_0^* and W_1^* are two challenged keywords. C chooses a random value $b \in \{0,1\}$ and uses W_b^* to compute a trapdoor Tw_b^* by running the trapdoor algorithm. The restrictions are that $IDi \neq IDc^*$ and $W_i \neq \{W_0^* \text{ or } W_1^*\}$. The Challenger then sends the trapdoor Tw_b^* to A

d) *Phase 2:* This phase is same as one defined in Game 1.

e) *Guess:* Finally, the adversary A output a guess of a value $b' \in \{0,1\}$. A wins the game if $b' = b$.

From the game 2 above, we define the advantage of the adversary A in breaking the trapdoor indistinguishability as the probability that the adversary A wins.

Definition 4: An IDERMKS scheme is said to have satisfied trapdoor indistinquishability against adaptive chosen plaintext attack if no adversary have a non-negligible advantage in Game 2.

5. THE PROPOSED IDERMKS SCHEME AND ITS SECURITY ANALYSIS

5.1 The Proposed IDERMKS Scheme

The proposed scheme consist of seven algorithms: System Setup, Key Extract, IDERMKS, Trapdoor Generation, Test, Ranking and Retrieval. Details of scheme is outlined below:

a) Setup: Given l as a security parameter, a trusted Private Key Generator (PKG) runs the setup algorithm as follows in order to generate a master private key and a master public key. Let G_1 be an additive cyclic group generated by P with a prime order of q and Let G_2 be a multiplicative cyclic group with same prime order q. The PKG generate a bilinear map $e: G_1 \times G_1 \rightarrow G_2$. The PKG randomly chooses a master secret key $x \in Z_q^*$ and compute the system public key P_{pub} as $P_{pub} = x.P$. Let H_1 , H_2 and H_3 be three cryptographic hash function such that $H_1: \{0,1\}^* \rightarrow G_1^*$, $H_2: \{0,1\}^* \rightarrow G_1^*$ and $H_3: G_2 \rightarrow \{0,1\}^n$ where n is a fixed length depending on l. The algorithm then publishes the system's parameters

$\{G_1, G_2, e, H_1, H_2, H_3, P, P_{pub}, n, params\}$ and the master secret keys x is kept secret.

b) *Key Extract*: Given a user's identity $ID_i \in \{0,1\}^*$, the PKG computes user's private key as $sk_{ID_i} \leftarrow xQ_{ID_i}$ where $Q_{ID_i} \leftarrow H_1(ID_i)$.

c) *IDERMKS*: The data owner chooses a random value $a \in Z_q^*$ and compute the searchable index on a set of keywords for a document as $I = (U, V)$ where $U = aQ_{ID}$ and $V = H_3(k)$ and k is computed as $k = e(U, P_{pub})e(H_2(w_i), P)$.

d) *Trapdoor Generation:* Let W be the set of keywords that the data user want to search for. The data user compute $t = H_2(W)$ and sends it to the data owner. The data owner then generate the trapdoor $T_w = ask_{ID} + t$ and sends it to the data user.

e) *Test*: Given the IDERMKS ciphertext index I and the trapdoor T_w, the cloud server runs the test algorithm to check if $H_3(e(T_w, P) = V$ then it will return file and it ID.

f) *Ranking:* We adopt a privacy preserving ranked search scheme implement in [18] into our IDERMKS scheme to facilitate the ranking of the candidate file to determine which file is more relevant to a certain keyword according to the encoded relevance scores. In this ranking scheme, the cloud server makes a comparison of the encoded relevant scores of the files without knowing the actual contents of the files.

g) *Retrieval:* Given the Identity Based Encryption of the symmetric key $C = (C_1, C_2)$ where $C_1 = rP$ and $C_2 = sk.(e(P_{pub}, Q_{ID})^r)$ where $r \in Z_q^*$ used to encrypt the document. The data user blind C by computing $C' = (C)^\alpha = (C_1^\alpha, C_2^\alpha)$ and sends it to the data owner. The data owner returns $sk^\alpha = C_2^\alpha.e(r\alpha P_{pub}, Q_{ID})^{-1}$ to the data user. The secret key can be deduced thereafter and be used to decrypt the encrypted document [23].

5.2 Security Analysis of the Proposed IDERMKS Scheme

We prove our IDERMKS system is a non-interactive searchable encryption scheme that is semantically secure in a random oracle model [20][21][22]. The proof of security relies on the difficulty of the BDH and CDH problem.

5.2.1 Ciphertext Indistinguishability

We demonstrate that the proposed IDERMKS scheme satisfies ciphertext indistinguishability under an adaptive chosen plaintext attack. For the security of our scheme to be simplified, we prove a lemma in which the adversary is assumed to be an outside attacker.

Lemma 5.2.1 *We assume that is an adversary A with a non-negligible probability ε_1 can break the ciphertext indistinguishability of the proposed IDERMKS scheme in the random oracle under an adaptive chosen plaintext attack, then there exist a challenger C with a non-negligible probability $\xi \geq \frac{2\varepsilon_1}{e(q_R+1).q_{H_3}}$ who can solve the BDH problem where q_R and q_{H_3} represents the maximum numbers of making key extract and H_3 queries respectively.*

Proof: The Challenger *C* is given an input of the Bilinear Diffie Hellman (BDH) parameters as $\{q, G_1, G_2, e\}$ produce by *G* and a random instances $(P, aP, bP, cP) = (P, P_1, P_2, P_3)$ of the BDH problem for these parameters, that is *P* is random in G_1^* and $a, b, c \in Z_q^*$ where q is the order of G_1 and G_2. Let $D = e(P, P)^{abc} \in G_2$ be the solution to the BDH problem. The Challenger *C* find D by interacting with the adversary *A* as follows:

a) *Setup:* The challenger *C* runs the setup algorithm to generate the public parameters $\{q, G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3\}$ by setting $Q_{ID} = P_2$ and $P_{pub} = P_3$. H_1, H_2 and H_3 are random oracles controlled by *C*. The challenger *C* gives the public parameters to *A*. The challenger *C* generate *A*'s private key sk_{ID_A} by running the Key Extract algorithm with the public key begin ID_A as $sk_{ID_A} = aQ_{ID} = abP$. The challenger keeps the secret key *msk* and *A*'s private key sk_{ID_A} is given to him.

b) *H_1 queries:* The adversary *A* can query the random oracle H_1 at any time in an adaptive manner. To respond to these queries, the Challenger *C* make a list of tuples $< ID_i, Q_i, b_i, coin_i >$, called the H_1^{LIST}. Until the adversary *A* makes queries to the oracle, H_1^{LIST} is initially empty. When the adversary *A* queries the oracle with ID_i the Challenger *C* responds as follows:

i. If query ID_i already exist on the H_1^{LIST} in the tuple $< ID_i, Q_i, b_i, coin_i >$ then C responds with $H_1(ID_i) = Q_i \in G_1^*$

ii. Otherwise C generates a random $coin_i \in \{0,1\}$ so that $\Pr[coin_i = 0] = \delta$ for some δ that will be determined later.

iii. The challenger C picks a random value $b_i \in Z_q^*$. If $coin_i = 0$, compute $Q_i = b_i P \in G_1^*$. If $coin_i = 1$, compute $Q_i = b_i Q_{ID} \in G_1^*$

iv. The challenger C adds the tuple $< ID_i, Q_i, b_i, coin_i >$ to H_1^{LIST} and respond to A with $H_1(ID_i) = Q_i$. Note that Q_i is uniform in G_1^* and is independent of A's current view as required.

c) *H_2 queries*: At any time, the adversary A can query the random oracle H_2. To respond to the queries, the Challenger C maintains a list of tuple $< ID_i, w_i, Qw_i, x_i >$ called the H_2^{LIST}. The list is initially empty until the adversary A makes a query. When the adversary A queries the oracle H_2 for $< ID_i, w_i >$, The challenger C respond as follows:

i. If $< ID_i, w_i >$ appears in the list H_2^{LIST}, C respond with $H_2(ID_i, w_i) = Qw_i$

ii. Otherwise, the challenger C randomly select a value $x_i \in Z_q^*$ and compute $Qw_i = H_2(ID_i, w_i) = x_i.P$. Finally, C adds the tuple $< ID_i, w_i, Qw_i, x_i >$ in the list H_2^{LIST} and respond to the adversary A with $H_2(ID_i, w_i) = Qw_i$

d) *H_3 queries:* The adversary A can query the random oracle H_3 at any time. To respond to the query, the challenger C maintains a list of tuple $< m_i, n_i >$ called H_3^{LIST} as described as follows: Until the adversary A queries the oracle H_3 for m_i, the list H_3^{LIST} is initially empty. When the adversary queries the oracle H_3, the challenger C responds as follows:

i. If m_i appear in the list H_3^{LIST}, the challenger C responds with $H_3(m_i) = n_i$

ii. Otherwise, the challenger C randomly select a value $n_i \in \{0,1\}^n$ and set $H_3(m_i) = n_i$. Finally, C add the pair $< m_i, n_i >$ to the H_3^{LIST} and responds to the adversary A with $H_3(m_i) = n_i$.

e) *Phase 1:* Let ID_i be a private key extraction query issued by A. The Challenger C responds as follows:

i. C runs the algorithm for responding to H_1-Queries to obtain a $Q_i \in G_1^*$ such that $H_1(ID_i) = Q_i$ and let $< ID_i, Q_i, b_i, coin_i >$ be the corresponding tuple on the H_1^{LIST}. If $coin_i = 1$, then C reports failure and abort. The attack on IDERMKS failed.

ii. If $coin_i = 0$, C gets $Q_i = b_i P$ and Define $sk_{ID_i} = b_i P_{pub} \in G_1^*$. Observe that $sk_{ID_i} = cQ_i$ and therefore sk_{ID_i} is the private key associated to public key ID_i. The Challenger then sends sk_{ID_i} to A.

f) *Trapdoor queries:* When the adversary A issues a query for the trapdoor for keyword $< ID_i, w_i >$, the Challenger C then access the corresponding tuples $< ID_i, Q_i, b_i, coin_i >$ and $< ID_i, w_i, Qw_i, x_i >$ in the H_1^{LIST} and H_2^{LIST} respectively and computes $T_{w_i} = x_i sk_{ID_i} + t$ where $x_i \in Z_q^*$ and $t = H_2(w_i)$ and returns T_{w_i} to the adversary A.

g) *Challenge:* The adversary A sends $(ID_C^*, W_0^* W_1^*)$ to the challenger C where W_0^* and W_1^* are two challenged keywords. Upon receiving $(ID_C^*, W_0^* W_1^*)$ from A, the challenger C chooses a random value $b \in \{0,1\}$ and access the corresponding tuple $< ID_c^*, W_b^*, Qw_b^*, x^* >$ in H_2^{LIST} to generate an IDERMKS ciphertext $I^* = (U^*, V^*) = (U^*, R)$, where $R \in \{0,1\}^n$ is a random value. The restrictions are that the adversary A did not make any private key extraction for IDc^* and A did not also make a trapdoor query for W_0^* and W_1^*. Finally, the challenger C sends I^* to the adversary A.

h) *Phase 2: A* can continue to make the key extract queries adaptively for any identity ID_C and the trapdoor query for any keyword W to the challenger C subject to the restriction that $IDc \neq IDc^*$ and $w \neq \{W_0^* \text{ or } W_1^*\}$.

i) *Guess:* Finally, the adversary *A* output a guess of a value $b' \in \{0,1\}$. *A* wins the game if $b' = b$.

By the assumption, *A* with a non-negligible probability ε_1 can distinguish the IDERMKS ciphertext I^* under an adaptive chosen plaintext attack. Now, the challenger *C* picks a tuple $< m^*, n^* >$ in the H_3^{LIST} and outputs $v^* = m^* / e(H_2(w_b), P)$ as the solution for the BDH instance (P, aP, bP, cP) for $a, b, c \in Z_q^*$. In the following, we demonstrate that the output $m^* / e(H_2(w_b), P)$ is equal to $e(P,P)^{abc}$ where $U = aQ_{ID}$ and $m^* = (e(U, P_{pub})e(H_2(w_b), P))$. The challenger *C* accesses the corresponding tuple $< ID_c^*, W_b^*, Qw_b^*, x^* >$ in the list H_3^{LIST} and computes

$$m_i^* / e(H_3(w_b), P) = e(U, P_{pub}).e(H_3(w_b), P) / e(H_3(w_b), P)$$
$$= (e(aQ_{ID}, P_{pub}).e(H_3(w_b), P) / e(H_3(w_b), P)$$
$$= (e(aQ_{ID}, cP) = e(abP, cP) = e(P,P)^{abc}$$

By adopting the similar technique used in [22][24], we compute the probability of our IDERMKS scheme by discussing the probability that the challenger *C* does not abort during the simulation. Suppose the adversary *A* makes q_R queries to the key extract query. In such an instance, the probability that the challenger *C* does not abort in phase 1 or 2 is $(\delta)^{q_R}$ and the probability that the challenger *C* does not abort during the challenged step is $(1-\delta)$. Therefore, the probability that the challenger *C* does not abort during the simulation is $(\delta)^{q_R} \cdot (1-\delta)^{q_R}$. This value can be maximize at $\delta_{abt} = 1 - \frac{1}{q_R + 1}$. By using δ_{abt}, the probability that the challenger *C* does not abort is at least $\frac{1}{e(q_R + 1)}$. We note that the probability analysis uses the same techniques as Coron's analysis of the Full Domain Hash in [25]. The Challenger *C* outputs the correct *D* with the probability at least $\frac{2\varepsilon_1}{qH_3}$ [25] where q_{H3} denotes the total number of making H_3 queries. Therefore, the Challenger *C* with a probability $\xi \geq \frac{2\varepsilon_1}{e(q_R+1).q_{H_3}}$ can solve the BDH problem. This contradict to the BDH assumption.
By Lemma 6.1, we obtain the following theorem.

Theorem 5.2.1: *The proposed IDERMKS scheme satisfies the ciphertext indistinquishability against an adaptive chosen plaintext attack under the BDH assumption in the random oracle.*

5.2.2 Trapdoor Indistinguishability

Lemma 5.2.2 *In the random oracle model, we assume that if there is an adversary A with a non-negligible advantage that can break the trapdoor indistinguishability of the proposed IDERMKS scheme under the adaptive chosen keyword attack, then there exist a challenger C with a non-negligible advantage who can solve the computational CBH problem.*

Proof: We assume that the challenger *C* receives a CDH instance of (P, aP, bP) for $a, b \in Z_q^*$ where q is the order of G_1 and G_2. By interacting with the adversary *A*, the challenger *C* will return the CDH solution abP in Game 2.

a) *Setup:* The Challenger C runs the setup algorithm to generate public parameters $\{q, G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3\}$ where $Q_{ID} = bP$ $P_{pub} = aP$. H_1 , H_2 and H_3 are random oracles controlled by C. The challenger C gives the public parameters to A. The Challenger C generate A's private key sk_{IDA} by running the Key Extract algorithm with the public key begin ID_A as $sk_{IDA} = xQ_{ID}$. The challenger C keeps the secret key msk and A's private key sk_{IDA} is given to him.

b) *H_1 queries:* The adversary A can query the random oracle H_1 at any time in an adaptive manner. To respond to these queries, the Challenger C make a list of tuples $< ID_i, Q_i, b_i, coin_i >$, called the H_1^{LIST}. Until the adversary A makes queries to the oracle, H_1^{LIST} is initially empty. When the adversary A queries the oracle with ID_i the Challenger C responds as follows:
   i. If query ID_i already exist on the H_1^{LIST} in the tuple $< ID_i, Q_i, b_i, coin_i >$ then C responds with $H_1(ID_i) = Q_i \in G_1^*$
   ii. Otherwise C generates a random $coin_i \in \{0,1\}$ so that $\Pr[coin_i = 0] = \delta$ for some δ that will be determined later.
   iii. The challenger C picks a random value $b_i \in Z_q^*$. If $coin_i = 0$, compute

$Q_i = b_i P \in G_1^*$. If $coin_i = 1$, compute $Q_i = b_i Q_{ID} \in G_1^*$

iv. The challenger C adds the tuple $< ID_i, Q_i, b_i, coin_i >$ to H_1^{LIST} and respond to A with $H_1(ID_i) = Q_i$. Note that Q_i is uniform in G_1^* and is independent of A's current view as required.

c) *H_2 queries:* At any time, the adversary A can query the random oracle H_2. To respond to the queries, the Challenger C maintains a list of tuple $< ID_i, w_i, Qw_i, x_i >$ called the H_2^{LIST}. The list is initially empty until the adversary A makes a query. When the adversary A queries the oracle H_2 for $< ID_i, w_i >$, The Challenger C respond as follows:

i. If $< ID_i, w_i >$ appears in the list H_2^{LIST}, C respond with $H_2(ID_i, w_i) = Qw_i$

ii. Otherwise, the challenger C randomly select a value $x_i \in Z_q^*$ and compute $Qw_i = H_2(ID_i, w_i) = x_i.P$. Finally, C adds the tuple $< ID_i, w_i, Qw_i, x_i >$ in the list H_2^{LIST} and respond to the adversary A with $H_2(ID_i, w_i) = Qw_i$

d) *H_3 queries:* The adversary A can query the random oracle H_3 at any time. To respond to the query, the challenger C maintains a list of tuple $< m_i, n_i >$ called H_3^{LIST} as described as follows: Until the adversary A queries the oracle H_3 for m_i, the list H_3^{LIST} is initially empty. When the adversary queries the oracle H_3, the challenger C responds as follows:

i. If m_i appear in the list H_3^{LIST}, the challenger C responds with $H_3(m_i) = n_i$

ii. Otherwise, the challenger C randomly select a value $n_i \in \{0,1\}^n$ and set $H_3(m_i) = n_i$. Finally, C add the pair $< m_i, n_i >$ to the H_3^{LIST} and responds to the adversary A with $H_3(m_i) = n_i$.

e) *Phase 1:* Let ID_i be a private key extraction query issued by A. The Challenger C responds as follows:

i. C runs the algorithm above for responding to H_1-Queries to obtain a $Q_i \in G_1^*$ such that $H_1(ID_i) = Q_i$ and let $< ID_i, Q_i, b_i, coin_i >$ be the corresponding tuple on the H_1^{LIST}. If $coin_i = 1$, then C reports failure and abort. The attack on IDERMKS failed.

ii. If $coin_i = 0$, C gets $Q_i = b_i P$ and Define $sk_{ID_i} = b_i P_{pub} \in G_1^*$. Observe that $sk_{ID_i} = cQ_i$ and therefore sk_{ID_i} is the private key associated to public key ID_i. The Challenger then sends sk_{ID_i} to A.

f) *Trapdoor queries:* When the adversary A issues a query for the trapdoor for keyword $< ID_i, w_i >$, the Challenger C then access the corresponding tuples $< ID_i, Q_i, b_i, coin_i >$ and $< ID_i, w_i, Qw_i, x_i >$ in the H_1^{LIST} and H_2^{LIST} respectively and computes $T_{w_i} = x_i sk_{ID_i} + t$ where $x_i \in Z_q^*$ and $t = H_2(w_i)$ and returns T_{w_i} to the adversary A.

g) *Challenge:* The adversary A sends $(ID_C^*, W_0^* W_1^*)$ to the challenger C where where W_0^* and W_1^* are two challenged keywords with the restrictions that $IDi \neq IDc^*$ and $W_i \neq \{W_0^* \text{ or } W_1^*\}$. The Challenger C then access the corresponding tuples $< ID_i, Q_i, b_i, coin_i >$ and $< ID_i, w_i, Qw_i, x_i >$ in the H_1^{LIST} and H_2^{LIST} respectively and computes $T_{wb}^* = x_i sk_{ID_i} + t$ where $x_i \in Z_q^*$ and $t = H_2(w_i)$ and finally C sends the trapdoor T_{wb}^* to the adversary A.

h) *Phase 2:* A can continue to make the key extract queries adaptively for any identity ID_C and the trapdoor query for any keyword W to the challenger C subject to the restriction that $IDc \neq IDc^*$ and $w \neq \{W_0^* \text{ or } W_1^*\}$.

i) *Guess:* Finally, the adversary A output a guess of a value $b' \in \{0,1\}$.

By assumption, the adversary *A* with a non-negligible advantage can distinguish the trapdoor T_{wb}^* under the adaptive chosen keyword attack. Meaning the trapdoor T_{wb}^* satisfy the equation $e(T_{wb}^*, P) = V'$ where $V' = e(U, P_{pub}).e(H_2(w_b), P)$ and let $U = xQ_{ID}$. We can then obtain:

$$e(T_{wb}^*, P) = e(U, P_{pub}).e(H_2(w_b), P)$$
$$e(T_{wb}^*, P) = e(xQ_{ID}, aP).e(H_2(w_b), P)$$
$$e(T_{wb}^*, P) = e(abP, P)^x.e(H_2(w_b), P)$$
$$e(T_{wb}^*, P) / e(H_2(w_b), P) = e(abP, P)^x$$
$$e(T_{wb}^* - H_2(w_b), P)^{x^{-1}} = e(abP, P)$$

Which implies that $e(abP, P) = e(T_{wb}^* - H_2(w_b), P)^{x^{-1}}$. Hence the Challenger can obtain $abP = x^{-1}(T_{wb}^* - H_2(w_b))$, which contradicts the CDH assumption.

Theorem 5.2.2 *The proposed IDERMKS scheme satisfies the trapdoor indistinquishability against an adaptive chosen plaintext attack under the CDH assumption in the random oracle.*

6 PRIVACY PRESERVING RANKED SEARCH

Our scheme adopts the privacy preserving ranked search used in [25] . It is important to note that, after the retrieval of all the candidate files, the cloud server cannot return all the undifferential files to the data user due to (1) acquisition of excessive communication cost and overhead for the system if the cloud server decides to return all the candidate files.(2) the data users may only be concerned with the top-k relevant files that correspond to their queries. The scheme in [25] illustration an additive order preserving and privacy preserving encoding scheme. It then uses the encoded relevance score to obtain the top-k search result.

7 PERFORMANCE ANALYSIS AND COMPARISON

In this section, we analyze the performance of the proposed IDERMKS scheme. Comparison will be made between the IDERMKS scheme and previously proposed PEKS schemes. In order to make it convenient in evaluating the computational cost of our IDERMKS scheme, our concentration will be on some time-consuming operations and by adopting similar strategy used in [22] and we define the time consuming operation as follows:

- TG_e: The execution time of a bilinear map operation $e: G_1 \times G_1 \to G_2$
- TG_{mul}: The execution time for scalar multiplication operation in G_1
- TG_H: The execution time for map-to-point hash function, thus $H_1, H_2, H_3: \{0,1\}^* \to G_1$
- T_{inv}: The execution time of a modular inverse operation in Z_q

The most time consuming operation is the time for executing a bilinear map operation TG_e as compared to the other operations stated above. In [26][27], the performance simulation results show that $TG_e \approx 2.5TG_{mul}$. We therefore analyze the performance for our IDERMKS scheme for each phase. In the IDERMKS ciphertext generation phase, it required $2TG_e + TG_{mul} + (n+1)TG_H$ to generate an IDERMKS ciphertext, where n represents the total number of keywords. In the trapdoor generation phase, $TG_{mul} + TG_H$ is required to generate a trapdoor T_w.

Table 1 list the comparison between our IDERMKS scheme and the previously proposed dPEKS schemes [22][28][29][30] in terms of public key setting and performance. From the table

TABLE 1: *Comparison between our IDERMKS and previously proposed dPEKS schemes*

	Scheme of Hwang and Lee [29]	**Scheme of Rhee et al.** [30]	**Scheme of Hu and Liu** [28]	**Scheme of Wu et al.** [22]	**Our IDERMKS**
Public Key Setting	Pairing-based	Pairing-based	Pairing-based	ID-based	ID-based
Certificate Management	Required	Required	Required	Not Required	Not Required
Computational cost for ciphertext generation (conjunctive *n* keywords)	$(2n+2)TG_{mul}$ $+2nTG_H$	$(2n+2)TG_{mul}$ $+2nTG_H$	$(2n+2)TG_{mul}$ $+2nTG_H$	$TG_e + (n+2)TG_{mul}$ $+(n+2)TG_H$	$2TG_e + TG_{mul}$ $+(n+1)TG_H$
Computational cost for ciphertext generation (1 keywords)	$3TG_{mul}$ $+2TG_H$	$TG_e + 2TG_{mul}$ $+TG_H$	$TG_e + 2TG_{mul}$ $+TG_H$	$TG_e + 3TG_{mul}$ $+3TG_H$	$2TG_e + TG_{mul}$ $+2TG_H$
Computational cost for trapdoor generation	$3TG_{mul} + 2TG_H$ $+T_{inv}$	$3TG_{mul} + 2TG_H$ $+T_{inv}$	$3TG_{mul} + 2TG_H$ $+T_{inv}$	$2TG_{mul} + TG_H$	$TG_{mul} + TG_H$
Computational cost for test	$3TG_e$	$TG_e + 2TG_{mul}$ $+TG_H$	$TG_e + 2TG_{mul}$ $+TG_H$	$2TG_e + TG_{mul}$ $+T_{inv}$	$TG_e + 2TG_{mul}$ $+T_{inv}$

in [22], it is easy to realize that the two schemes in [28][30] do not support conjunctive keywords since they require $nTG_e + (n+1)TG_{mul} + nTG_H$ in the generation of their ciphertexts. We categorically state that our scheme is more efficient in the ciphertext generation when *n* is sufficiently large. Furthermore, our scheme is more efficient in the trapdoor generation phase as compared to other dPEKS schemes. Also, our scheme is based on ID-system which eliminates the load of certificate management associated with the other schemes which are based on pairing-based public key system. The scheme in [22] is also based on ID-system but our IDERMKS is more efficient in terms of ciphertext and trapdoor generation.

8 CONCLUSION

In this paper we proposed an IDERMKS scheme which supports conjunctive keywords. We defined the framework and the security requirements for our IDERMKS scheme and when we compared our scheme to previous dPEKS scheme, the performance of our scheme is more efficient in both the ciphertext and trapdoor generation phase. Our ID-based system also has an advantage of eliminating certificate management associated with PKI. We also demonstrated that our IDERMKS scheme possesses the ciphertext indistinquishability and trapdoor indistinguishability under BDH and CDH assumptions respectively. Our future work will seek to implement our IDERMKS scheme in a hybrid cloud and also to investigate our IDERMKS scheme without random oracle with a pairing free algorithm to determine how feasible it will be in cloud.

REFERENCES

[1] S. Sagiroglu and D. Sinanc, "Big data: A review," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 42–47.

[2] I. Abaker, T. Hashem, I. Yaqoob, N. Badrul, S. Mokhtar, A. Gani, and S. Ullah, "The rise of ' big data ' on cloud computing : Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, 2015.

[3] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005, vol. 3783 LNCS, pp. 414–426.

[4] M. Albanese, S. Jajodia, R. Jhawar, and V. Piuri, "Secure Cloud Computing," *Secur. Cloud Comput.*, pp. 239–259, 2014.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," *Proc. - IEEE INFOCOM*, 2010.

[6] D. Wagner, A. Perrig, D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy*, 2000, pp. 44–55.

[7] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," *Appl. Cryptogr. Netw. Secur.*, vol. 3089, pp. 31–45, 2004.

[8] R. Li, Z. Xu, W. Kang, K. C. Yow, and C. Z. Xu, "Efficient multi-keyword ranked query over encrypted data in cloud computing," *Futur. Gener. Comput. Syst.*, vol. 30, no. 1, pp. 179–190, 2014.

[9] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," *2010 Proc. IEEE INFOCOM*, pp. 1–5, 2010.

[10] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proceedings - IEEE INFOCOM*, 2014, pp. 2112–2120.

[11] M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," in *Proceedings - International Conference on Distributed Computing Systems*, 2011, pp. 273–281.

[12] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Adv. Cryptol.*, vol. 196, pp. 47–53, 1985.

[13] D. O. F. Tampering, "On the," *Advances*, 1987.

[14] D. H??hnlein, M. J. Jacobson, and D. Weber, "Towards practical non-interactive public-key cryptosystems using non-maximal imaginary quadratic orders," *Des. Codes, Cryptogr.*, vol. 30, no. 3, pp. 281–299, 2003.

[15] U. Maurer and Y. Yacobi, "Non-interactive public-key cryptography," *Adv. Cryptology—EUROCRYPT'91*, pp. 498–507, 1991.

[16] L. Problem, "An ID-Based Cryptosystem Based on the Discrete," vol. 7, no. 4, 1989.

[17] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[18] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1566–1577, 2016.

[19] P. D. Mgmt, C. A. Secondary, and C. Author, "Distributed and Parallel Databases An Efficient Privacy-Preserving Multi-Keyword Search over Encrypted Cloud Data with."

[20] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *J. ACM*, vol. 51, no. 4, pp. 557–594, 2004.

[21] M. Bellare, "R a n d o m Oracles are Practical : A Paradigm for Designing Efficient P r o t o c o l s," pp. 62–73, 1993.

[22] T.-Y. Wu, T.-T. Tsai, and Y.-M. Tseng, "Efficient searchable ID-based encryption with a designated

server," *Ann. Telecommun. - Ann. Des Télécommunications*, vol. 69, no. 7–8, pp. 391–402, 2014.

[23] C. Orencik and E. Savas, "Efficient and secure ranked multi-keyword search on encrypted cloud data," *Proc. 2012 Jt. EDBT/ICDT Work.*, pp. 186–195, 2012.

[24] Y. M. Tseng and T. T. Tsai, "Efficient revocable ID-based encryption with a public channel," *Comput. J.*, vol. 55, no. 4, pp. 475–486, 2012.

[25] J.-S. Coron, "On the Exact Security of Full Domain Hash," *CRYPTO 2000 Adv. Cryptol.*, pp. 229–235, 2000.

[26] T.-Y. Wu and Y.-M. Tseng, "An ID-Based Mutual Authentication and Key Exchange Protocol for Low-Power Mobile Devices," *Comput. J.*, vol. 53, no. 7, pp. 1062–1070, 2009.

[27] T.-Y. Wu and Y.-M. Tseng, "An efficient user authentication and key exchange protocol for mobile client–server environment," *Comput. Networks*, vol. 54, no. 9, pp. 1520–1530, 2010.

[28] C. Hu and P. Liu, "An enhanced searchable public key encryption scheme with a designated tester and its extensions," *J. Comput.*, vol. 7, no. 3, pp. 716–723, 2012.

[29] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4575 LNCS, pp. 2–22, 2007.

[30] 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," *J. Syst. Softw.*, vol. 83, no. 5, pp. 763–771, 2010.