British Journal of Mathematics & Computer Science
16(6): 1-11, 2016, Article no.BJMCS.25804
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Symmetric Key Encryption with Conjunctive Field Free Keyword Search Scheme

Sher Ali Fairouz[1,2*] and Song Feng Lu[1]

[1] *Huazhong University of Science and Technology, Wuhan, Hubei 430074, P.R.China.*
[2] *Kufa University, Kufa, Iraq.*

Authors' contributions

This work was carried out in collaboration between both authors. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJMCS/2016/25804
Editor(s):
(1) Victor Carvalho, Polytechnic Institute of Cvado and Ave, Portuguese Catholic University and Lusiada University, Portugal.
Reviewers:
(1) S. K. Rososhek, Tomsk State University, Russia.
(2) Xiaoqiang Zhang, China University of Mining and Technology, China.
(3) Vipin Saxena, Babasaheb Bhimrao Ambedkar University, India.
(4) S. Geetha, VIT University, Chennai, India.
Complete Peer review History: http://sciencedomain.org/review-history/14600

Received: 21st March 2016
Accepted: 27th April 2016
Published: 12th May 2016

Original Research Article

Abstract

Searchable symmetric encryption (SSE) enables a sender to outsource the encrypted files to a cloud server, while maintaining the ability to conditionally search over it without knowing the sensitive data contents. Prior work in this area has focused on single keyword search. Conjunctive Keyword Searches (CKS) schemes improve system usability by retrieving the matched files, but in this type of search the receiver has to provide the server with a trapdoor for every individual keyword in the conjunction and rely on an intersection protocol to retrieve the correct set of files, in other words the receiver has to repeatedly perform the search protocol for many times based on the number of keywords in the conjunction. Most of the existing (CKS) schemes use conjunctive keyword searches with fixed position keyword fields, this type of search is not applicable for many real applications, such as the database text and the body of e-mail. In this paper, we propose a

*Corresponding author: E-mail: *fairouz_sherali@yahoo.com*;

new trapdoor-indistinguishable symmetric key encryption with conjunctive keyword search, which does not need to specify the positions of the keywords. Instead of giving the server one trapdoor for each keyword in the conjunction set, the recipient give it a trapdoor for multiple keywords in the conjunction set to search on encrypted files. Furthermore, the search process could not reveal any information about the number of keywords in the query expression. Our proposed scheme is proven secure against chosen-keyword attacks under the Decisional Diffie-Hellman(DDH) assumption in the random oracle model.

Keywords: Searchable encryption; symmetric key encryption; conjunctive keyword search; keyword field free.

1 Introduction

The proliferation of Cloud computing enables mobile clients to access their data from anywhere and at any time. More and more cloud services have spread all around the world such as computing resource, storage space outsourcing and different kinds of software applications. For reasons of low cost, efficiency, convenience, better connectivity and etc., clients often store their data on remote servers. Since more remote servers are public, there exist a lot of risks for the data during the process of data transfer, clients ensure the privacy of their data by storing it in encrypted image, then they can search the encrypted data and retrieve it. The first effort of searching encrypted data by keyword was tackled by Song, Wagner and Perrig [1]. To securely search through encrypted data, searchable encryption schemes have been proposed in recent years, which can be divided into two types: symmetric searchable encryption (SSE) [2, 3, 4, 5, 6, 7, 8, 1], and asymmetric searchable encryption (ASE)[9, 10, 11, 12, 13, 14, 15, 16, 17]. To perform a search on a dataset, a client creates an index of keywords listed in the files and later on executes the search on the index in a way that allows the server to retrieve the files contain a certain keyword instead of retrieving all the encrypted files back which is fully impractical solution in cloud computing scenarios.

Most classical searchable encryption works emphases only on single keyword search [10, 4, 5, 6, 1] or multiple keyword search [18, 19, 7, 14, 15]. In the symmetric key schemes, recently some solutions have been proposed for general Boolean queries on encrypted data [20, 21], and there are only two related schemes in the public key setting [22, 23].

There are many Boolean operations, like conjunction, disjunction and negation. In the conjunctive search the client can search for the encrypted files containing: w_1 and w_2 and w_n. While in the disjunctive search the client can search for encrypted files containing: w_1 or w_2 or w_n, and finally in the negative search the client can search for all encrypted files which do not contain particular words.

To enhance search functionalities, many boolean keyword search works over encrypted data have been proposed. Obviously, there are two trivial solutions to achieve conjunctive keyword search: the first is to get the intersection of all sets of files where each set is the searching result for every keyword in the conjunctive; the second is to define a meta-keyword for every possible conjunction of keywords . Existing schemes for conjunctive keywords search ([7] and subsequent works) were supporting keyword fields in the index. This setting is not useful and much more difficult to search in most systems, such as the database text and the body of e-mail.

In our paper, we define a secure scheme of symmetric key encryption with keyword field free conjunctive keyword searches (SSE-KFF-CKS) that allows conjunctive keyword search queries on encrypted data without needing to specify the positions of the keywords (hide the keyword positions from the querier) where the keywords can be in any arbitrary order. Furthermore, instead of giving the server a trapdoor for each keyword in the conjunction, we combine individual keywords to make them regarded as one keyword. To do so, we use the template concatenation function to concatenate the keywords in the conjunction as $w_1\|w_2\|...\|w_m$ without needing to conjunctive search mark $\wedge$ (see more details in Section 2.4). In another meaning if the clients want to retrieve the files that contain a set of keywords, they should not repeat the search protocol for m keywords times. Also, we illustrate that our work is secure against adaptive Chosen-Keyword Attacks (CKA) in the random oracle model ROM under the Decisional Diffie Hellman(DDH) assumption.

1.1 Main contributions

Our main contributions can be summarized as:

1. Our scheme dealing with keyword field-free conjunctive keyword searches, we design a novel algorithm that converts the conjunctive keywords search to a single keyword search and consequently the model cannot support the posting list intersection protocol. With this new scheme, we can greatly reduce the search time.
2. Security of the proposed scheme based on the Decisional Diffie-Hellman (DDH) assumption. This scheme states that the remote server should learn nothing, especially the number of keywords in conjunction set, other than the result of the conjunctive query.

1.2 Previous work

The first symmetric key schemes for keyword search via encrypted data are introduced in [1]. The authors consider a setting in which the owner of file encrypts each word of a file separately. Goh[6] proposed a method for secure index using Bloom filters and introduced the notion of semantic security against adaptive chosen-keyword attacks. Determining whether a file contains a keyword can be done securely in constant time. In the public key works, Boneh et al.[10] first introduced public key scheme for keyword search, where anyone can use public key and write to the data stored on remote server, but only authorized recipients with the secret key can search. An efficient implementation of a public key work for keyword search specifically designed for documents that are the audit trails of clients querying a database is in [16]. However, these above works emphasis only on single keyword search.

Conjunctions in the Searchable Symmetric Encryption setting were first proposed by Golle et al.[7]. Their works consisted of two schemes: the first scheme compares two hash codes of the keywords to find the required files. This scheme is based on DDH assumption and proven secure in the ROM model. The transmission cost of the trapdoors is very high. The second one, tests two outputs of bilinear pairing constructed from input keywords and checks if the keywords are included in the file. This scheme achieves a constant size of trapdoors, but its security analysis relies on a nonstandard model. Boneh and Waters[13] developed a Public key Encryption with Keyword Search scheme[10] for conjunctive keyword searches from a generalization of Anonymous Identity-Based Encryption[9]. This scheme supports equality, comparison(such as greater-than) and general subset queries. The storage and communication costs are linearly dependent on the number of fields. Baek et al. [24]have addressed important problems in removing secure channel, refreshing keywords, and processing multiple keywords, which have not been tackled in the original PEKS scheme. Byun et al.[25] suggested an efficient conjunctive keyword search scheme using a number of pairings operations, this scheme requires constant communication and storage costs. Moreover, the scheme is more efficient than both schemes by Golle et al.[7] in term of communication overhead, but it has a higher computational overhead of the encryption process for each file by requiring as many pairing operations as number of keyword fields. Ryu and Takagi [26]introduced an efficient scheme for CKS where the size of the trapdoors for several keywords is nearly the same as for a single keyword. The authors use asymmetric pairings in groups of prime order. The encryption process requires one pairing per file and the server has to perform two pairings per file to search. Hwang and Lee[14] also introduced a public key encryption scheme with the conjunctive keyword search (PECK)based on bilinear map and gave a new concept called multiuser PECKS which is the first model for multi-user public key encryption with the conjunctive keyword search (mPECKS) scheme. The notion of this scheme is to reduce the communication and storage overhead for the cloud server and also for the client. Kerschbaum[27] proposed a searchable encryption scheme with conjunctive keyword search without specifying the position of keywords.

Recently, Wang et al. [28]proposed the first keyword-field-free conjunctive keyword search scheme. The security of this scheme is based on the l-decisional Diffie-Hellman inversion and discrete logarithm assumptions. The notion is to remove the keyword fields by using a bilinear map per keyword per file index. Furthermore, the authors extend their scheme for dynamic groups and prove its security under the Weak Diffie-Hellman assumption and LRSW assumption.

1.3 Security requirements

a. Data security [17]: when the senders encrypt the keywords and the message by the authorized receiver's public key, only the corresponding secret key can decrypt the content of the file, that mean no one could derive the embedded keywords from the ciphertext.

b. Client authentication: after encrypting, no information can be extracted from the trapdoor and the ciphertexts, but the cloud server still has to check whether the clients who send the trapdoor are the authorized clients. [29, 30, 31].

c. Trapdoor security [17]: whenever the receiver wants to search the encrypted data, he sends the trapdoor containing the corresponding keywords to the cloud server; other clients can get nothing from the trapdoor even if the trapdoors are obtained by the attackers.

d. Against off-line keyword-guessing attack: any proposed security model should stand against outside attackers and inside attackers (malicious servers)[19, 14].

1.4 Outline

The rest of our paper is organized as follows. Section 2 gives the outline of the proposed scheme, preliminaries, notations, semantic security of the SSE-KFF-CKS scheme and construction of SSE-KFF-CKS. Section 3 introduces the security analysis. Finally, Section 4 provides brief conclusions.

2 Outline of the Proposed Scheme

2.1 Preliminaries

We briefly show theoretical background and complexity assumptions that used throughout our paper.

2.1.1 Bilinear pairing

We say a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map if the following properties hold:

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of the same prime order q and $\hat{e}(g, g)$ is efficiently computable;
- For all $x, y \in Z_q$ and $g \in \mathbb{G}_1$, then $\hat{e}(g^x, g^y) = \hat{e}(g, g)^{xy}$;
- $\hat{e}(g, g)$ is non-degenerate. That is, if g generates $\mathbb{G}_1$ the $\hat{e}(g, g)$ generates $\mathbb{G}_2$.

The above bilinear map is called symmetric pairings.

2.1.2 Decisional Diffie-Hellman (DDH) assumption

We say that the decisional Diffie-Hellman (DDH) problem is hard if, for any PPT distinguisher $\mathcal{A}$, the function

$|Pr[\mathcal{A}(\mathbb{G}_1, q, g, g^x, g^y, g^z) = 1] - Pr[\mathcal{A}(\mathbb{G}_1, q, g, g^x, g^y, g^{xy}) = 1]|$, is negligible.

2.2 Notations

- F: the collection of n plaintext file to be outsourced, denoted as $F = \{F_1, F_2, ..., F_n\}$.
- ID: the collection of n files identifiers, denoted as $ID = \{ID_1, ID_2, ..., ID_n\}$.
- Enc_F: the collection of k retrieved files from the remote server contained the conjunctive keyword, denoted as $Enc_F = \{Enc_{F_1}, Enc_{F_2}, ..., Enc_{F_k}\}$, where $Enc_F \subseteq F$.
- $\mathcal{W}_{Fi}$: the collection of m distinct keywords, m is relatively small, per trapdoor extracted from each file F_i in collection F, denoted as $\mathcal{W}_{Fi} = \{w_1, w_2, ..., w_m\}$.
- P_{F_i}: the collection of possible permutation extracted from keywords sequence $\mathcal{W}_{Fi}$, denoted as $P_{F_i} = \{pr_1, pr_2, ..., pr_{m!}\}$
- Pr_j: the collection of m keywords regards as one keyword using concatenation operation, denoted as $pr_j = \{w_1 \| w_2 \| ... \| w_m\}$, $j = 1...m!$.
- Q: the collection of l queries in a search request, denoted as $Q = \{q_1, q_2, ..., q_l\}$.
- Tq: the trapdoor for l conjunctive query denoted as $Tq = \{q_1 \| q_2 \| ... \| q_l\}$.

2.3 Semantic security of the SSE-KFF-CKS scheme

The proposed scheme is semantically secure (indistinguishability) against a chosen keyword attack IND-CKA if every PPT (Probabilistic Polynomial Time) attacker has a negligible advantage.

Given the security parameter σ, the challenger $\mathcal{C}$ runs the key generation algorithm $KeyGen(\sigma)$ to generate secret key SK and keep it to itself.

Let $\mathcal{A}$ be an attacker that can adaptively ask $\mathcal{C}$ for the trapdoor T_W for any keyword $\mathcal{W} \in \{0,1\}^*$ of it's choice, where $\mathcal{W} = \{w_1 || w_2 || ... || w_s\}$.

$\mathcal{A}$ selects two sets of conjunctive word $\mathcal{W}_0 = \{w_{01} || w_{02} || ... || w_{0s}\}$ and $\mathcal{W}_1 = \{w_{11} || w_{12} || ... || w_{1s}\}$, which are not to be asked any trapdoors previously, then sends them to the challenger. After that $\mathcal{C}$ selects a random element $\eta \in \{0,1\}$ and sends the attacker the encryption of conjunctive keyword $\mathcal{W}_\eta$.

$\mathcal{A}$ can continue to ask for trapdoors T_W for any conjunction of keywords $\mathcal{W} = \{w_1 || w_2 || ... || w_s\}$ of his choice as long as $\mathcal{W} \neq \mathcal{W}_0, \mathcal{W}_1$.

Eventually, $\mathcal{A}$ outputs a guess $\eta' \in \{0,1\}$ and wins the game if $\eta = \eta'$.

$\mathcal{A}'s$ advantage in breaking SSE-KFF-CKS scheme is defined as:

$$|Adv_{\mathcal{A}}(\sigma) = |Pr[\eta = \eta'] - \frac{1}{2}|.$$

2.4 Construction

In our scheme we do not target the fixed field keyword like Golle *et al.* scheme[7]; we rather consider an improved query model consisting of Boolean expression on keywords expressed in the conjunctive form without needing to specify the positions of the keywords where the keywords can be in any arbitrary order.

In our model, we have the sender S, the receiver R and the cloud server. Let F be a file collection consisting of n files, where ID_i is a unique file identifier. S extracts m keywords from each file F_i as $\mathcal{W}_{F_i} = \{w_1, w_2, ..., w_m\}$ and combines them as one keyword with the different $m!$ possible permutations of conjunctive keyword P_{F_i}=$\{pr_1, pr_2, ..., pr_{m!}\}$, where each permutation set pr_j has m combined keywords, $Pr_j = \{w_1||w_2||...||w_m\}$ where $j \in [1, m!]$. For example, if $m = 3$ keywords, and the keywords are P,Q,R. The Sender creates 6 different permutations of the such keywords sequence. Each permutation set, consists of three keywords, regards as one keyword using concatenation operation P_{F_i}=$\{(P\|Q\|R), (P\|R\|Q), (Q\|P\|R), (Q\|R\|P), (R\|P\|Q), (R\|Q\|P)\}$.

When the receiver wants to retrieve the file ID_i that has, e.g, the following keywords (P and Q and R), he combines such individual keywords to make them regarded as one query, then he can create a one trapdoor as one search token and sends it to the server. In other meaning R can send one of the following conjunctive keyword $(P\|Q\|R), (P\|R\|Q), (Q\|P\|R), (Q\|R\|P), (R\|P\|Q)$ or $(R\|Q\|P)$ as a query to the cloud server. Then the server tests the conjunctive keyword P_{F_i} against the trapdoor and retrieves the associated matched file to the R without needing for the posting list intersection protocol.

Our scheme consists of six algorithms *KeyGen, FilEncrypt, KeyEncrypt, TrapdoorGen,*
Search, FilDecrypt which are scattered between two phases, Setup Phase and Retrieval Phase.

2.4.1 Setup phase

This phase includes three algorithms as detailed below:

I. *KeyGen*: The sender S initiates the scheme by using $\mathsf{KeyGen}(\sigma)$ algorithm. This algorithm takes the security parameter σ as input to create the following parameters: q as a large prime number, two groups $\mathbb{G}_1$, $\mathbb{G}_2$ of order q, g is a random generator of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$, e is a random element of $\mathbb{Z}_q^*$ and one cryptographic hash functions $H : \{0, 1\}^* \rightarrow \mathbb{G}_1$.

II *FilEncrypt*: To protect data privacy and undesired accesses, the file collection F should be encrypted before outsourcing them onto remote servers which are not within their trusted domains. To do so, S encrypts each file $F_i \in F$ using AES algorithm. Each file F_i comprising of a unique identifier $ID_i \in \{0, 1\}^n$. To protect the file identifiers ID_i, S encrypts this ID_i also with AES encryption technique, such technique assurances that if the same file identifier is encrypted multiple times, it will create different ciphertexts but all decrypted to the same value.

III *KeyEncrypt*: S extracts the conjunctive keyword $\mathcal{W}_{F_i}$ from each file $F_i \in F$ and encrypts them. To do so, the sender creates $m!$ possible permutations set of these keywords sequence $P_{F_i} = \{pr_1, pr_2, ..., pr_{m!}\}$ and makes each permutation pr_j looks like one keyword using concatenation operation as $pr_j = \{w_1\|w_2\|...\|w_m\}$ where $j \in [1, m!]$, then he chooses a random number $f \in \mathbb{Z}_q^*$. Finally the algorithm $\mathsf{KeyEncrypt}$ returns C_j for each permutation Pr_j as follows:

$$C_j = (X, Y),$$

where $X = g^f$ and $Y = \hat{e}(H(pr_j)^e, g^f)$

The final step in the setup phase algorithm is sending the C_j and encrypted files to remote server.

2.4.2 Retrieval phase

Include three algorithms as detailed below:

I- *TrapdoorGen*: To retrieve only the files containing the conjunctive query $Q = \{q_1, q_2, ..., q_l\}$, the receiver R chooses a random number $k \in \mathbb{Z}_q^*$, then he creates one trapdoor for a conjunction of queries Q. To do so, the receiver combines the conjunctive queries to make them look like one query, $Tq = \{q_1 \| q_2 \| ... \| q_l\}$, then R will compute the trapdoor of the search request of concatenated conjunctive keywords as follows:

$$Tw = (U, V)$$

where $U = g^k$ and $V = g^k H(Tq)^e$.

Finally, R submits Tw to the cloud server.

II- *Search*: Upon receiving the trapdoor Tw, server will call the Search algorithm on each conjunctive query, this algorithm will check whether the following equality holds:

$$\frac{Y.\hat{e}(U,X)}{\hat{e}(V,X)} = \frac{\hat{e}(H(pr_j)^e, g^f).\hat{e}(g^k, g^f)}{\hat{e}(g^k H(Tq)^e, g^f)} = \frac{\hat{e}(H(pr_j), g)^{ef}\hat{e}(g,g)^{kf}}{\hat{e}(H(Tq), g)^{ef}\hat{e}(g,g)^{kf}} = 1$$

If so, the server returns the relevant encrypted file corresponding the ID_i to R. Otherwise, it returns no files.

III- *FilDecrypt*: Once R receives the encrypted files from cloud server, he calls FilDecrypt algorithm to decrypt each retrieved file $Enc_{F_i} \in Enc_F$ using the AES encryption technique algorithm.

3 Security Analysis

Theorem 3.1. *The proposed scheme SSE-KFF-CKS is semantically secure against chosen-keyword attacks in the RO model under the Decisional Diffie Hellman assumption*

Proof. Suppose there is an attack algorithm $\mathcal{A}$ that has advantage ϵ in breaking our scheme. Suppose $\mathcal{A}$ makes q_H hash queries to H and q_T trapdoor queries. Then we built an algorithm $\mathcal{C}$ that solves the DDH problem with the advantage at least $\epsilon' = \epsilon/e((m!)q_T + 1)$ where *m!* is the number of possible permutations of conjunctive keyword.

Algorithm $\mathcal{C}$ is given an instance $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, g^x, g^y, g^z)$, where *x,y,z* are random elements in Z_q. It's goal is to decide whether $z = xy$.

- **KeyGen**: $\mathcal{C}$ chooses a random element e as it's own secret key, then sets $\big(e = y\big)$ and $\big(log_g^{H(\mathcal{W})} = x\big)$.
  Hash queries. To respond to H queries, $\mathcal{C}$ maintains a list of tuples $\langle \mathcal{W}_j, h_j, d_j \rangle$ called the H-list. The list is initially empty. When the attacker issues a hash query for a conjunctive keyword $\mathcal{W}_i = \{w_1||w_2||...||w_s\}$, algorithm $\mathcal{C}$ checks whether $\mathcal{W}_i = \mathcal{W}_j$, if so, algorithm $\mathcal{C}$ answers consistently with the previous queries by responding with $H(\mathcal{W}_i) = h_i$. Otherwise, $\mathcal{C}$ generates a random coin $d_i \in \{0, 1\}$ so that $Pr[d_i = 0] = 1/(q_T + 1)$, then $\mathcal{C}$ selects a random element $\gamma_i \in Z_q$, if $d_i = 0$, $\mathcal{C}$ computes $h_i = g^{\gamma_i}$, otherwise, $\mathcal{C}$ computes $h_i = g^x$, $\mathcal{C}$ adds the tuple $\langle \mathcal{W}_i, h_i, d_i \rangle$ to H-list, and responds to $\mathcal{A}$ with $H(\mathcal{W}_i) = h_i$.
  When $\mathcal{A}$ requests an encryption of conjunctive keyword $\mathcal{W}$, algorithm $\mathcal{C}$ calls the above algorithm for responding to H-queries to obtain an $h_i \in \mathbb{G}_1$. Then he searches the H-list

$\langle \mathcal{W}_i, h_i, d_i \rangle$ for conjunctive keyword $\mathcal{W}_i$, if $d_i = 1$ then $\mathcal{C}$ aborts. Otherwise, we know that $d_i = 0$ hence $H(\mathcal{W}_i) = g^{\gamma_i}$, then $\mathcal{C}$ computes

$$C = \left(g^f, \hat{e}(H(\mathcal{W}_i)^e, g^f)\right) = \left(g^f, \hat{e}((g^{\gamma_i})^e, g^f)\right) = \left(g^f, \hat{e}((g^y)^{\gamma_i}, g^f)\right)$$

- **Trapdoor queries**. When the attacker issues a query for conjunctive keyword $\mathcal{W}_i$, Algorithm $\mathcal{C}$ calls the above algorithm for responding to H-queries to obtain an $h_i \in \mathbb{G}_1$, then searches the H-list $\langle \mathcal{W}_i, h_i, d_i \rangle$ for conjunctive query. if $d_i = 1$ then $\mathcal{C}$ aborts. Otherwise, we know that $d_i = 0$ hence $H(\mathcal{W}_i) = g^{\gamma_i}$, then $\mathcal{C}$ computes

$$Tw = \left(g^k, g^k H(\mathcal{W}_i)^e\right) = \left(g^k, g^k (g^{\gamma_i})^e\right) = \left(g^k, g^k (g^x)^{\gamma_i}\right)$$

- **Challenge**. Algorithm $\mathcal{A}$ chooses and sends two conjunctive keyword $\mathcal{W}_0 = \{w_{01}||w_{02}||...||w_{0s}\}$ and $\mathcal{W}_1 = \{w_{11}||w_{12}||...||w_{1s}\}$ to $\mathcal{C}$, and $\mathcal{A}$ must not have asked previously for the trapdoors of $\mathcal{W}_i$ where $i \in \{0, 1\}$. For each conjunctive keyword $\mathcal{W}_i$, algorithm $\mathcal{C}$ calls the random oracle algorithm for responding to H-queries to get $h_0, h_1 \in \mathbb{G}_1$, where $H(\mathcal{W}_i) = h_i$. Algorithm $\mathcal{C}$ randomly chooses a $\eta \in \{0, 1\}$, if $d_\eta = 0$ then $\mathcal{C}$ aborts. Otherwise, we know that $d_\eta = 1$ hence $H(\mathcal{W}_i) = g^x$, then $\mathcal{C}$ computes

$$Ch = \left(g^f, \hat{e}(H(\mathcal{W}_\eta)^e, g^f)\right) = \left(g^f, \hat{e}((g^x)^y, g^f)\right) = \left(g^f, \hat{e}(g^z, g^f)\right)$$

- **More queries.** After the above challenge query, $\mathcal{A}$ can perform additional trapdoor queries with same restriction that $\mathcal{W}_i \neq \mathcal{W}_0, \mathcal{W}_1$, $\mathcal{C}$ answers these queries as before.
- **Output.** $\mathcal{A}$ outputs its guess $\eta' \in \{0, 1\}$. If $\eta' = \eta$, $\mathcal{C}$ outputs that $z = xy$, otherwise, $\mathcal{C}$ replies $z \neq xy$.

To complete the proof of theorem 3.1, we now use the same technique as in [10] to analyze the probability that $\mathcal{C}$ does not abort during the above experiment. We define the following two events:

- $Event_1$: $\mathcal{C}$ does not abort during the Trapdoor queries.
- $Event_2$: $\mathcal{C}$ does not abort during the Challenge queries.

We suppose that both events $Event_1$ and $Event_2$ occur with sufficiently high probability. Let us consider the first event $Event_1$, the probability of $Event_1$ is $(1 - 1/(m!q_T + 1))^{m!q_T} \geq 1/e$, where $1/(m!q_T + 1)$ is the probability that a trapdoor query makes $\mathcal{C}$ to abort.

For the second event $Event_2$, the algorithm $\mathcal{C}$ does not abort during the challenge phase if one of d_0 and d_1 is 0. By the definition of H-list $Pr[d_\eta = 0] = 1/(m!q_T+1)$ where $\eta \in \{0, 1\}$ and the two values are independent of one another, we have that both $Pr[d_0 = d_1 = 1] = 1 - 1/q_T \geq (1 - 1/(q_T + 1))^2$. Hence, the $Pr[Event_2]$ is at least $1/q_T$. Consequently, the probability that $\mathcal{C}$ does not abort during the entire simulation is $Pr[Event_1 \wedge Event_2] \geq 1/(eq_T)$.

As a result, if the advantage of $\mathcal{A}$ against the proposed scheme is ϵ , the success probability of the algorithm $\mathcal{C}$ against the DDH challenge is at least $\epsilon/(e(m!q_T + 1)$. □

3.1 Comparisons

We compare our scheme to the previous conjunctive keyword schemes in terms of security assumption with other attributes in Table 1. The table is arranged by the query expressiveness. The first column shows the paper and reference. The "*security assumption*" column shows the security definitions, assumptions, and whether ROM is used to prove the secure of the scheme. The "*keyword field free*" column shows whether the scheme uses the fixed-position keyword fields keyword search or the keyword field free keyword search. The "*using in unstructured data*" column shows whether the scheme is practical for using in unstructured data or not. The "*Index generation*" column shows whether the construction of each scheme is based on the index generation or not. The last column shows whether the schemes can be used with a single or multi user.

Table 1. Comparison of security assumption and other attributes

Scheme	Security Assumption	Keyword filed free	Using in unstructured data	Index generation	User
Golle et al.-I [7]	IND1-CKA under DDH in the ROM	×	×	-	single user
Golle et al. -II [7]	IND1-CKA under new nonstandard hardness assumption	×	×	-	single user
Ballard et al. [18]	IND1-CKA based on the security of SSS in the ST	×	×	uses a pseudo-random function per keyword	single user
Byun et al. [25]	IND1-CKA under BDH in ROM	×	×	-	single user
Wang et al. [28]	IND1-CKAt under DL, 1-DDHI in ST	✓	✓	uses l-degree polynomial per document and compute l hash functions	multi user
Ryu et al. [26]	IND1-CKA under coXDH in ROM	×	×	-	single user
Cash et al. [20]	IND2-CKA under DDH	×	✓	uses pseudo - random function and hash function	multi user
Our scheme	IND1-CKA under DDH in ROM	✓	✓	-	single user

4 Conclusions

Briefly the proposed scheme focused on the enhancement of search options on encrypted data. We proposed a scheme of SSE-KFF-CKS scheme for cloud storage services which enables a client with one trapdoor of multiple keywords(conjunctive string) to search on encrypted data. Our construction is based on the symmetric key encryption with the conjunctive keyword search. The traditional schemes which only allowed searching for single keywords take *O(m)* for communication cost and $O(n^m)$ search time for m keywords. While the proposed scheme can greatly reduce the search time, because the clients should not repeat the search protocol for m keywords times, that mean the scheme requires just $O(1)$ for communication cost and *O(n)* search time for all keywords in conjunctive string. Furthermore, the server searches the encrypted files efficiently without leaking any information about the number of keywords in the conjunctive query. Compared with other schemes, our construction is more efficient and practical when applied to a cloud environment especially for unstructured text. Finally, we proved that the scheme is secure against adaptive chosen-keyword attacks in ROM under the Decisional Diffie Hellman Assumption.

Acknowledgement

The author would like to thank professor Song Feng Lu (School of Computer Science and Technology, Huazhong University of Science and Technology) for giving advice on this work and improving this paper, and to express many thanks to the referees for their valuable suggestions.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Song D, Wagner D, Perrig A. Practical techniques for searches on encrypted data. Proc. of IEEE Symposium on Security and Privacy; 2000.

[2] Bao F, Deng R, Ding X, Yang Y. Private query on encrypted data in multi-user settings. Proc. of ISPEC; 2008.

[3] Chase M, Kamara S. Structured encryption and controlled disclosure. In ASIACRYPT, LNCS. Springer, Dec. 2010;577-594.

[4] Chang YC, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data. Proc. of ACNS; 2005.

[5] Curtmola R, Garay JA, Kamara S, Ostrovsky R. Searchable symmetric encryption: Improved definitions and efficient constructions. Proc. of ACM CCS; 2006.

[6] Goh EJ. Secure indexes. 2003;216.
Cryptology ePrint Archive: http://eprint.iacr. org/

[7] Golle P, Staddon J, Waters B. Secure conjunctive keyword search over encrypted data. In: Jakobsson M, Yung M. (eds.) ACNS 2004. LNCS. Springer, Heidelberg. 2004;3089:31-45.

[8] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption. In Proc. of CCS; 2012.

[9] Abdalla M, Bellare M, Catalano D, Kiltz E, Kohno T, Lange T, Malone-Lee J, Neven G, Paillier P, Shi H. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In V. Shoup, editor, CRYPTO, Springer. 2005;3621 of LNCS:205-222.

[10] Boneh D, Crescenzo GD, Ostrovsky R, Persiano G. Public key encryption with keyword search. Proc. of EUROCRYP; 2004.

[11] Boneh D, Boyen X. Efficient selective identity-based encryption without random oracles. Journal of Cryptology. 2010;24:659-693.

[12] Boneh D, Franklin M. Identity-based encryption from the Weil pairing. Advances in Cryptology CRYPTO. 2001;213-229.

[13] Boneh D, Waters B. Conjunctive, subset, and range queries on encrypted data. Proc. of TCC. 2007;535-554.

[14] Hwang YH, Lee PJ. Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: Takagi et al. (eds.) Pairing. LNCS. Springer, Heidelberg. 2007;4575:2-22.

[15] Park D J, Kim K, Lee PJ. Public key encryption with conjunctive field keyword search. In: Lim CH, Yung M. (eds.) WISA Springer, Heidelberg. 2004;3325:73-86.

[16] Waters B, Balfanz D, Durfee G, Smetters D. Building an encrypted and searchable audit log. Proc. of 11th Annual Network and Distributed System; 2004.

[17] Zhang BO, Zhang F. An efficient public key encryption with conjunctive-subset keywords search. Journal of Network and Computer Application. 2011;34(1):262-267.

[18] Ballard L, Kamara S, Monrose F. Achieving efficient conjunctive keyword searches over encrypted data. In: Qing et al. (eds.) ICICS. LCS, Springer, Heidelberg. 2005;3783:414-426.

[19] Chen Z, Wu C, Wang D, Li S. Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor. In: Chau et al. (eds.) PAISI. LNCS, Springer, Heidelberg. 2012;7299:176-189.

[20] Cash D, Jarecki S, Jutla CS, Krawczyk H, Rosu M, Steiner M. Highly- scalable searchable symmetric encryption with support for boolean queries. In Canetti R, Garay J. (eds.) CRYPTO, Springer, Heidelberg, LNCS. 2013;8042:353-373.

[21] Moataz T, Shikfa A. Boolean symmetric searchable encryption. In: ACM ASIACCS. 2013;265-276.

[22] Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart N. (ed.) EUROCRYPT. LNCS. Springer, Heidelberg. 2008;4965:146-162.

[23] Lai J, Zhou X, Deng RH, Li Y, Chen K. Expressive search on encrypted data. In: ACM ASIACCS. 2013;243-252.

[24] Baek J, Safavi-Naini R, Susilo W. Public key encryption with keyword search revisited. In: Proceedings of Computational Science and Its Applications, ICCSA, LNCS, Springer-Verlag. 2008;5072:1249-1259.

[25] Byun J, Lee D, Lim J. Efficient conjunctive keyword search on encrypted data storage system. Proceedings of Euro PKI, LNCS, Springer-Verlag. 2006;4043:184-196.

[26] Ryu EK, Takagi T. Efficient conjunctive keyword-searchable encryption. In AINAW. IEEE Computer Society, Washington, DC. 2007;409-414.
DOI: http://dx.doi.org/10.1109/AINAW.166

[27] Kerschbaum F. secure conjunctive keyword searches for unstructured text. 5th International Conference on Network and System Security (NSS). 2011;285-289.

[28] Wang P, Wang H, Pieprzyk J. Keyword field-free conjunctive keyword searches on encrypted data and extension for dynamic groups. In CANS (LNCS), Springer. 2008;5339:178-195.

[29] Kumar M. A new secure remote user authentication scheme with smart cards. International Journal of Network Security. 2010;11(2):88-93.

[30] Lee CC. On security of an efficient nonce-based authentication scheme for SIP. International Journal of Network Security. 2009;9:201-203.

[31] Tsai CS, Lee CC, Hwang MS. Pass-word authentication schemes: Current status and key issues. International Journal of Network Security. 2006;3(2):101-115.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/14600