3-2018

VMKDO: Verifiable multi-keyword search over encrypted cloud data for dynamic data-owner

Yibin MIAO

Jianfeng MA

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Zhiquan LIU

Limin SHEN

See next page for additional authors

DOI: https://doi.org/10.1007/s12083-016-0487-7

Published in Peer-to-Peer Networking and Applications, 15 August 2016, Pages 1-11. Advance Online
http://doi.org/10.1007/s12083-016-0487-7

VMKDO: Verifiable multi-keyword search over encrypted cloud data for dynamic data-owner

Yinbin Miao[1] · Jianfeng Ma[1] · Ximeng Liu[2] · Zhiquan Liu[1] · Limin Shen[1] · Fushan Wei[3]

Abstract The advantages of cloud computing encourage individuals and enterprises to outsource their local data storage and computation to cloud server, however, data security and privacy concerns seriously hinder the practicability of cloud storage. Although searchable encryption (SE) technique enables cloud server to provide fundamental encrypted data retrieval services for data-owners, equipping with a result verification mechanism is still of prime importance in practice as semi-trusted cloud server may return incorrect search results. Besides, single keyword search inevitably incurs many irrelevant results which result in waste of bandwidth and computation resources. In this paper, we are among the first to tackle the problems of data-owner updating and result verification simultaneously. To this end, we devise an efficient cryptographic primitive called as verifiable multi-keyword search over encrypted cloud data for dynamic data-owner scheme to protect both data confidentiality and integrity. Rigorous security analysis proves that our scheme is secure against keyword guessing attack (KGA) in standard model. As a further contribution, the empirical experiments over real-world dataset show that our scheme is efficient and feasible in practical applications.

Keywords Cloud storage · Searchable encryption · Result verification · Data-owner updating · Keyword guessing attack

Jianfeng Ma
jfma@mail.xidian.edu.cn

[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China

[2] School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore, Singapore

[3] State Key Laboratory of Mathematical Engineering and Advanced Computing, The PLA Information Engineering University, Zhengzhou, China

1 Introduction

As the fundamental component of cloud computing [1, 2], cloud storage [3] offers an opportunity for considerable number of enterprises and individuals to reduce the heavy burden of local data computations and managements. However, a large number of sensitive data (such as financial documents, personal emails, etc.) is now placed on the semi-honest-but-curious cloud service provider (CSP) which may compromise data privacy. Though encryption is a straightforward and efficient way to eliminate the data security and privacy concerns against semi-trusted CSP, it makes search over encrypted data extremely difficult. The typical solution to tackle this problem is SE technique [4, 5] which allows data-owners to retrieve encrypted files according to user-specified keywords. For the sake of saving bandwidth and computing resources, SE schemes should support multi-keyword search to avoid returning irrelevant encrypted files in practice.

In principle, CSP should ensure data confidentiality and integrity according to specified protocols. However, data-owners actually move their computing tasks to the semi-trusted cloud server which may return incorrect results to

save computational resources or maintain its reputation. Therefore, result verification mechanism [6] should be furnished to guarantee the correctness of search results. Moreover, the computational costs of result verification should be as small as possible in order not to cancel out the advantages of cloud storage.

In practical situations, the search right of certain data-owner may be shifted to other data-owner without leaking private key, as shown in Fig. 1. More specifically, we consider a scenario in which certain data-owner (Data-owner A, for example, a doctor in charge of patient medical records) has been revoked from trusted domain. To issue multi-keyword search over encrypted data and gain correct search results, new data-owner (Data-owner B, as the delegatee of Data-owner A) enables CSP to convert a small part of original ciphertext into another form which can be searched by himself through proxy re-encryption technique. Besides, he empowers a private audit server to check the correctness of search results. More importantly, ciphertext updating and result verification should not incur heavy computational burden for resource-limited entities, especially for mobile terminals and sensor nodes.

To tackle aforementioned problems, we extend public audit technique [7, 8] to SE scheme, and then devise an efficient cryptographic primitive called as **V**erifiable **M**ulti-**k**eyword Search over Encrypted Cloud Data with Dynamic **D**ata-**O**wner (VMKDO) scheme to achieve multi-keyword search and result verification simultaneously. Note that the multi-keyword search (including conjunctive keyword search and disconjunctive keyword search) in our scheme just supports conjunctive keyword search. Specifically, our main contribution can be summarized as follows:

1) ***Multi-keyword search***. Our scheme enables data-owners to issue multiple keywords in a search query.
2) ***Result verification***. With the result verification mechanism, our scheme can prevent CSP from returning false or inaccurate search results.

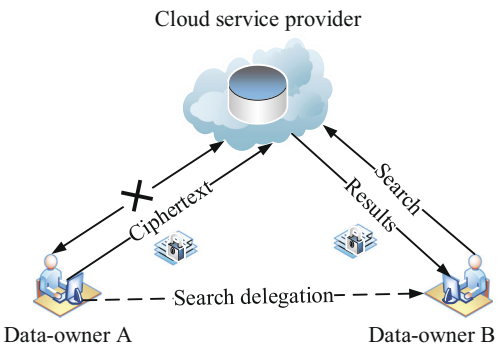

Fig. 1 Scenario in proposed scheme

3) ***Data-owner updating***. With proxy re-encryption technique our scheme can support dynamic data-owner by updating a small part of original ciphertext.
4) ***Security and efficiency***. The formal security analysis proves that our scheme can resist KGA in the standard model, and experimental results over a real-world dataset show its efficiency in practice.

The remaining of this paper is organized as follows. Section 2 first introduces the previous work associated with our scheme. The preliminaries are presented in Section 3, followed by Section 4 which gives the system model, threat model and design goals. Then Section 5 demonstrates the concrete construction of our scheme in detail. Section 6 shows the correctness, security and performance analysis. Finally, the concluding remark of this whole paper is summarized in Section 7.

2 Related work

To the best of our knowledge, SE which enables data-owners to securely search over ciphertext through keywords and selectively retrieve files of interest has drawn much attention in both industrial and academic fields. And existing SE schemes can be roughly divided into two categories, namely symmetric SE and asymmetric SE. Since Song et al. [9] proposed the first symmetric SE scheme and Boneh et al. [4] presented the first asymmetric SE scheme, considerable number of SE schemes enriched with various functionalities [10–13] have been further researched.

Multi-keyword search Although subsequent SE schemes [10, 14, 15] have enhanced the security and improved the efficiency, these scheme are still limited to single keyword search. To shrink the searching scope over encrypted data and retrieve exact results quickly, SE schemes should support multi-keyword search [11, 16–20] instead of results intersection. As the first multi-keyword search scheme proposed by Golle et al. [21] just supported general queries (equality search), Boneh et al. [17] presented a more practical scheme which supported arbitrary conjunctive queries (such as comparison query, subset query, etc.). And Hwang et al. [18] came up with a secure multi-keyword search scheme in the asymmetric setting and extended it to multi-user system to admit a broad range of applications.

Result verification search However, in practice, CSP may execute parts of search operations and returns a fraction of search results to save computation and bandwidth resources, thereby leading to integrity violation. To tackle this problem, Chai et al. [6] gave the first verifiable keyword search

scheme in symmetric setting. Aiming to overcome the limitation in single user setting, Zheng et al. [22] and Sun et al. [23] presented the fine-grained keyword search schemes through utilizing attribute-based encryption [24–26]. Whereas, the aforementioned schemes were still confined to single keyword search. To overcome this defect, Sun et al. [27] constructed a verifiable multi-keyword search scheme over dynamic encrypted data to support file collection update (such as insertion, deletion, etc.), Miao et al. [28] presented a verifiable multi-keyword search scheme by removing the secure channel. Whereas, these schemes cannot be applied in the dynamic data-owner setting.

Proxy re-encryption with keyword search In some scenarios, certain data-owner may be revoked from the trusted domain or delegate his search right to delegatee (new data-owner), while updating the whole ciphertext inevitably incurs heavy computational burden. To the best of our knowledge, proxy re-encryption technique [29–32] can convert the original ciphertext encrypted by old data-owner into new form that can be accessed by new data-owner. Though Guo et al. [32] introduced the notion of searchable proxy re-encryption scheme with a designated tester, this scheme just supports single keyword search. Along this direction, Yang et al. [33] demonstrated a more secure multi-keyword search scheme in the standard model through proxy re-encryption technique, but there still existed a limitation in this scheme which it couldn't guarantee the accuracy of search results.

To enrich the search functionalities over encrypted data, our scheme can achieve aforementioned functionalities (as illustrated in Table 1) simultaneously.

3 Preliminaries

Given a set S, the symbol $s \in_R S$ is defined as choosing an element s uniformly at random from the set S. Then we simply review some cryptographic background through the following definitions.

Table 1 Functionality comparison

Schemes	MKS	SRV	DOU
VABKS [22]	×	✓	×
ABKS-UR [23]	×	✓	✓
VCKS [27]	✓	✓	×
Re-dPEKS [32]	×	×	✓
Re-dtPECK [33]	✓	×	✓
VMKDO	✓	✓	✓

– "MKS": Multi-keyword Search;
– "SRV": Search Result Verification;
– "DOU": Data-Owner Updating.

Definition 1 (Bilinear map) Let G_1, G_2 be two multiplicative cyclic groups of prime order p, g be a generator of group G_1, and e be the bilinear map $G_1 \times G_1 \rightarrow G_2$ with following properties:

(1) Bilinearity: Given four elements $a, b \in_R G_1, u, v \in_R Z_p^*$, we can have $e(a^u, b^v) = e(a^v, b^u) = e(a, b)^{uv}$.
(2) Non-degeneracy: $e(g, g) \neq 1$.
(3) Computability: Given elements $a, b \in_R G_1$, there exits an efficient algorithm to compute $e(a, b)$.

Definition 2 (Discrete Logarithm (DL) Assumption) Let G_1 be a group of order p, and g be the generator of G_1. For any probabilistic polynomial time adversary $\mathcal{A}$, its advantage on solving the DL problem in group G_1 is negligible, which is defined as $Pr[\mathcal{A}(g, g^a) = a] \leq \epsilon$, where $a \in_R Z_p^*$.

Definition 3 (Decisional Bilinear Diffie-Hellman (DBDH) Problem) Let (G_1, G_2, p, g, e) be the bilinear map parameters. Given the tuple (g^a, g^b, g^c, Z), the DBDH problem is to decide whether Z equals to $e(g, g)^{abc}$ or to a random element in G_2. Where $a, b, c \in_R G_1, Z \in_R G_2$.

Definition 4 (Truncated Decisional q-Augmented Bilinear Diffie-Hellman Exponent (q-ABDHE) Problem) Let (G_1, G_2, p, g, g', e) be the bilinear map parameters, where (g, g') are both the generators of G_1. Given the tuple $(g', g'_{q+2}, g, g_1, \cdots, g_q, Z)$, the truncated decisional q-ABDHE problem is to decide whether Z equals to $e(g', g_{q+1})$ or an element in G_2. Where $g_i = g^{a^i} (1 \leq i \leq q+1)$, $g'_{q+2} = g'^{a^{q+2}}$, $a \in_R Z_p^*$, $Z \in_R G_2$.

4 Problem formulation

Let $[1, n]$ be a series of integer set $\{1, 2, ..., n\}$, an integer k be the security level, and $(\mathcal{F}, \mathcal{W})$ be the file and keyword space, respectively. Besides, search token and trapdoor will be used interchangeably throughout this paper.

4.1 System model

The cloud storage system considered in this paper involves three main entities (as shown in Fig. 2), namely cloud service provider (CSP), private audit server (PAS) and data-owner (DO). Where DO uploads ciphertext (indexes and signatures) to CSP and can issue search queries when necessary, CSP provides data storage and retrieval services for DO, PAS is responsible for verifying the correctness of search results. When DO wants to conduct a search query, he needs to submit a search token to CSP, then CSP matches it with indexes and returns the relevant encrypted files to

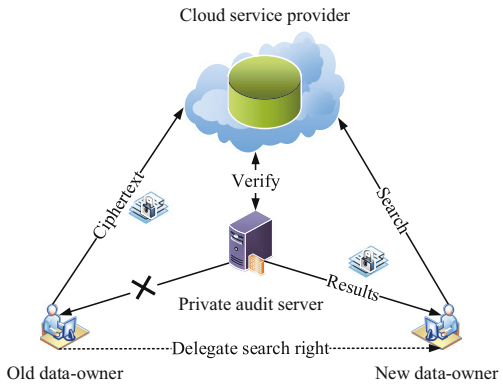

Fig. 2 System model of our scheme

PAS. Once the search results pass the result verification, PAS sends the search results to DO. Besides, DO can delegate his search right to other DO without leaking private key when he leaves or is revoked.

4.2 Threat model

In this paper, PAS is assumed to be a trusted entity and honestly checks whether the search results are correct or not. Like most of previous SE schemes, CSP is still considered to be semi-honest-but-curious. Specifically, CSP will honestly perform the pre-defined protocols, but it is curious to execute parts of search operations, even return forged or false search results under various motivations. While for the DO and his delegatee, they are both authorized entities at first. Once DO leaves or is revoked from the trusted domain, he cannot access the sensitive information. To enable delegatee to issue search queries, the original ciphertext should be updated.

4.3 Design goals

To enable secure search over encrypted data, our scheme should realize the following design goals.

1) ***Ciphertext updating***. When certain DO leaves or is revoked, our scheme should allow him to delegate his search right to other DO without leaking secret key.
2) ***Multi-keyword search***. To accurately locate the required encrypted files, our scheme should enable data-owners to issue multiple keywords search at the same time.
3) ***Security goals***. As the keyword set is always selected from a small space, and the security in random oracle has its own inherent problems, our scheme should resist KGA in the standard model. Besides, result verification mechanism should be provided to ensure data integrity.
4) ***Efficiency and feasibility***. To gain a broad range of applications and not incur extra computational burden during the ciphertext updating and result verification processes, our scheme should be efficient and feasible in practice.

5 Proposed VMKDO scheme

In this section, we first formally present the definition of our scheme, then give the concrete construction of our scheme.

5.1 Solution framework and security model

Our scheme is a tuple of seven algorithms including **Setup**, **KeyGen**, **ReKey**, **Enc**, **Trap**, **Search** and **Verify**, and these algorithms are presented as follows:

1) **Setup**$(1^k)\rightarrow$ $\{\mathcal{GP}, PK, SK\}$: Given the security parameter k, this deterministic algorithm outputs the global parameters $\mathcal{GP}$ and the public/secret key pair (PK, SK) of the traditional public key encryption algorithm.
2) **KeyGen**$(\mathcal{GP})\rightarrow$ $\{PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i}, PK_s, SK_s\}$. Perform this probabilistic algorithm to output the public/secret key pairs $\{(PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i}), (PK_s, SK_s)\}$ for certain $DO_i \in \mathcal{DO}$ and CSP, respectively. Where $\mathcal{DO}$ is denoted as the authorized DO-list.
3) **ReKey**$(SK_{\mathcal{O}_i}, SK_{\mathcal{O}_j})\rightarrow$ $\{rk_{i\rightarrow j}\}$: CSP performs the probabilistic algorithm to generate the re-encryption key $rk_{i\rightarrow j}$.
4) **Enc**$(\mathcal{GP}, F, W, PK, PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i}, PK_s, rk_{i\rightarrow j})\rightarrow$ $\{Sig_i, I_i, \pi\}$. DO_i first conducts this probabilistic algorithm to generate the signature set Sig_i, index set I_i and auxiliary information π, then he sends them to CSP. When DO_i delegates his search right to DO_j, CSP updates the signature set and a small part of index set through re-encryption key $rk_{i\rightarrow j}$.
5) **Trap**$(\mathcal{GP}, SK_{\mathcal{O}_i}, W', L)\rightarrow$ $\{T\}$: DO_i first runs this probabilistic algorithm to generate the search token T for queried keyword set W', then he sends T and the location set L to CSP.
6) **Search**$(\mathcal{GP}, T, L, I_i, SK_s)\rightarrow$ $\{C', ID'\}$: According to the queried location set L, CSP issues this deterministic algorithm to return relevant encrypted file set C' and corresponding identity set ID' to PAS if and only if the search token T matches with the index set I_i.
7) **Verify**$(\mathcal{GP}, PK_{\mathcal{O}_i}, C', ID')\rightarrow$ $\{0, 1\}$: PAS runs this algorithm to check the correctness of search results C' through initiating interactions with CSP. If C' passes the result verification, PAS returns it to DO_i. Otherwise, it aborts the results. Where "0" means that C' is incorrect, "1" means that C' is correct.

As the size of keyword space $\mathcal{W}$ is limit, most of previous SE schemes cannot resist dictionary attack and off-line keyword guessing attack. To this end, we utilize a designated tester [34–36] to issue test algorithm to avoid keyword guessing attack. Like the Re-dPEKS scheme [32], our scheme considers the adversary $\mathcal{A}$ to be either a malicious CSP or DO in Game 1 and Game 2, respectively.

Definition 5 (**Security model**) Let an integer k be the security parameter and $\mathcal{A}$ be a polynomial-time attacker, then we show the Game 1, Game 2 between $\mathcal{A}$ and simulator $\mathcal{B}$ in the following.

First, we assume that $\mathcal{A}$ is a malicious CSP, then we show the Game 1 as follows:

- **Init**: $\mathcal{B}$ first calls **Setup** and **KeyGen** algorithms to output the global parameters $\mathcal{GP}$, public/secret key pairs $\{(PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i}), (PK_s, SK_s)\}$ for certain DO_i and CSP, respectively. Then he sends the tuple $\{\mathcal{GP}, PK_{\mathcal{O}_i}, PK_s, SK_s\}$ to $\mathcal{A}$.
- **Search token queries 1**: $\mathcal{A}$ adaptively issues a number of search queries for distinct keyword set $\{W_1', \cdots, W_q'\}$ to the trapdoor generation oracle:
    1) **Trap oracle**: $\mathcal{B}$ first runs **Trap** algorithm to generate the search tokens $\{T_{W_i'}\}(1 \le i \le q)$, then he sends them to $\mathcal{A}$.
- **Challenge**: $\mathcal{A}$ first submits two target keyword sets (W_0^*, W_1^*) to be challenged on, then $\mathcal{B}$ selects a random bit $b \in \{0, 1\}$ and issues the **Enc** algorithm to generate the target ciphertex I_b^*. Finally, he sends it to $\mathcal{A}$.
- **Search token queries 2**: $\mathcal{A}$ issues a number of search token queries as in **Search token queries 1**, the only restriction is that the two keyword sets (W_0^*, W_1^*) cannot be queried to **Trap oracle**.
- **Guess**: $\mathcal{A}$ returns a guess bit $b' \in \{0, 1\}$ and wins this game if $b' = b$.

The $\mathcal{A}$'s advantage in breaking Game 1 is defined as $Adv_{\mathcal{A}}^{Game\ 1}(1^k) = 2Pr[b' = b] - 1$.

Second, let $\mathcal{A}$ be an outside attacker (such as the revoked DO), and we show the Game 2 as follows:

- **Init**: $\mathcal{B}$ first outputs the global parameters $\mathcal{GP}$, public/secret key pairs $\{(PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i}), (PK_s, SK_s)\}$, for certain DO_i and CSP, respectively. Then he sends the tuple $\{\mathcal{GP}, PK_{\mathcal{O}_i}, PK_s, SK_s\}$ to $\mathcal{A}$.
- **Challenge**: $\mathcal{A}$ first outputs two target keyword sets (W_0^*, W_1^*) to be challenged on. Once gaining this, then $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and calls the **Enc** algorithm to create a target index I_b^*. Finally he sends it to $\mathcal{A}$.
- **Guess**: $\mathcal{A}$ outputs his guess bit $b' \in \{0, 1\}$ and wins this game on the condition that $b' = b$.

Table 2 Notation descriptions

Notations	Descriptions
$F = \{f_1, \cdots, f_n\}$	File set
$ID = \{id_1, \cdots, id_n\}$	Identity set
$C = \{c_1, \cdots, c_n\}$	Ciphertext set
$Sig_i = \{sig_{i,1}, \cdots, sig_{i,n}\}$	DO_i' signature set
$W = \{w_1, \cdots, w_m\}$	Keyword set
$I_i = \{I_{i,1}, \cdots, I_{i,n}\}$	DO_i' index set
$\pi = \{\pi_0, \pi_1, \pi_2\}$	Auxiliary information
$W' = \{w_1', \cdots, w_l'\}$	Queried keyword set
$L = \{L_1, \cdots, L_l\}$	Location set of W' in W
$T = \{T_1, T_2\}$	Trapdoor for W'
$C' = \{c_1', \cdots, c_d'\}$	Search results
$ID' = \{id_1', \cdots, id_d'\}$	Returned identity set
$\{r, \tau_r\}(1 \le r \le d)$	Challenging information
(η, σ)	Proof information

The $\mathcal{A}$'s advantage in breaking Game 2 is defined as $Adv_{\mathcal{A}}^{Game\ 2}(1^k) = 2Pr[b' = b] - 1$.

Then we say that our scheme is secure against KGA in Game 1 and Game 2 when our scheme's advantage $Adv_{\mathcal{A},VMKDO}^{KGA}(1^k) = Adv_{\mathcal{A}}^{Game\ i}(1^k)$ $(i \in \{1, 2\}$ in resisting KGA is negligible.

5.2 Concrete construction

Before giving the specific construction of our scheme, we summarize some notations used in this paper in Table 2. In this system, as the files are encrypted by the traditional public key encryption algorithm, which is beyond the scope of our discussion. Thus the following algorithms mainly focus on building index and generating signatures on encrypted files.

Setup(1^k) On input the security parameter k, this deterministic algorithm first outputs the bilinear map parameters $(G_1, G_2, e, p, g_1, g_2)$, where G_1, G_2 are two groups of order p, $e : G_1 \times G_1 \to G_2$ is the bilinear map, and g_1, g_2 are two generators of G_1. Then it selects two hash functions $H_1 : \{0, 1\}^* \to_R G_1$, $H_2 : \{0, 1\}^* \to_R Z_p^*$ and returns the public/secret key pair (PK, SK) of public key encryption algorithm. Finally, it publishes the global parameters $\mathcal{GP}$ through Eq. 1, where PK is used to encrypt files, and SK shared among authorized DOs can decrypt encrypted files.

$$\mathcal{GP} = \{G_1, G_2, e, p, g_1, g_2, H_1, H_2, PK\}. \quad (1)$$

KeyGen($\mathcal{GP}, \mathcal{DO}$) Assume $\mathcal{DO}$ be the authorized DO set. For each $DO_i \in \mathcal{DO}$, this probabilistic algorithm first

selects two elements $b_i \in_R Z_p^*, \beta_i \in_R G_1$ and computes $B_i = g_1^{b_i}$. For CSP, this algorithm then chooses two elements $a \in_R Z_p^*, \alpha \in_R G_1$ and computes $A = g_1^a$. Finally this algorithm defines the the public/secret key pairs $(PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i})$, (PK_s, SK_s) of DO_i and CSP by Eq. 2, respectively.

$$\begin{aligned} &PK_{\mathcal{O}_i} = (B_i, \beta_i), SK_{\mathcal{O}_i} = b_i; \\ &PK_s = (A, \alpha), SK_s = a. \end{aligned} \tag{2}$$

ReKey($SK_{\mathcal{O}_i} = b_i, SK_{\mathcal{O}_j} = b_j$) When DO_i leaves and delegates his search right to DO_j, CSP first selects an element $\varepsilon \in_R Z_p^*$ and sends it to DO_i. Then DO_i computes ε / b_i and sends it to DO_j, and DO_j returns $b_j \varepsilon / b_i$ to CSP. Finally CSP generates the re-encryption key $rk_{i \to j} = b_j / b_i$.

Enc($\mathcal{GP}, F, W, PK, PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i}, PK_s, rk_{i \to j}$) The DO_i runs this probabilistic algorithm to generate the signatures and indexes for file set F according to keyword set W, which is shown in Fig. 3.

- **Step 1**: Given the file set F, this algorithm encrypts it as C through the traditional public key encryption algorithm. For each encrypted file $c_s \in C (1 \le s \le n)$ with identity id_s, DO_i generates the signature $sig_{i,s}$ through Eq. 3

$$sig_{i,s} = (H_1(id_s) g_2^{H_2(c_s)})^{b_i}. \tag{3}$$

- **Step 2**: Given the keyword set W, DO_i builds index for each file $f_s \in F$. He first chooses two elements $\lambda, \mu \in_R Z_p^*$, then he computes $\pi_0 = g_1^{\lambda}, \pi_1 = \nu \cdot e(g_1, g_1)^{\mu}, \pi_2 = e(g_1, \beta_i)^{\mu}$ and sets the index $I_{i,s}$ through Eq. 4, where $\nu = e(A, \alpha)^{\lambda}$.

$$I_{i,s} = \{I_0, I_t\} (1 \le t \le m), I_0 = B_i^{\mu}, I_t = g_1^{-w_t \mu}. \tag{4}$$

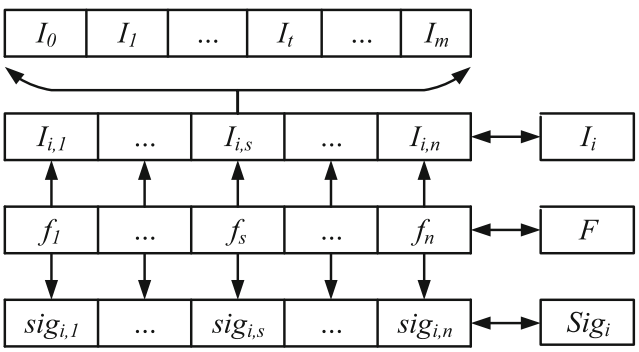

Fig. 3 Process of **Enc** algorithm

- **Step 3**: Finally, DO_i sends the signature set Sig_i, encrypted index set I_i and auxiliary information π to CSP, where Sig_i, I_i, π are defined by Eq. 5.

$$\begin{aligned} &Sig_i = \{sig_{i,1}, \cdots, sig_{i,n}\}, \\ &I_i = \{I_{i,1}, \cdots, I_{i,n}\}, \pi = \{\pi_0, \pi_1, \pi_2\}. \end{aligned} \tag{5}$$

- **Step 4**: When DO_i leaves and delegates search right to DO_j, CSP just needs to update the signatures and a small part of indexes through Eq. 6, where $1 \le s \le n, 1 \le t \le m$.

$$\begin{aligned} &sig_{j,s} = sig_{i,s}^{rk_{i \to j}} = (H_1(id_s) g_2^{H_2(c_s)})^{b_i}; \\ &I_{j,s} = \{I_0', I_t'\}, I_0' = I_0^{rk_{i \to j}} = B_j^{\mu}, I_t' = I_t. \end{aligned} \tag{6}$$

Trap($\mathcal{GP}, SK_{\mathcal{O}_i}, W', L$) DO_i runs this probabilistic algorithm to generate search token for queried keyword set $W' = \{w_1', \cdots, w_l'\}$. He first selects an element $T_1 = \theta \in_R Z_p^*$ and sets $T_1 = \theta$, then he computes T_2 through Eq. 7.

$$T_2 = (\beta_i g_1^{-\theta})^{1/(b_i - \sum_{k=1}^{l} w_k')}. \tag{7}$$

Finally he sends the search token $T = \{T_1, T_2\}$ and the location set $L = \{L_1, \cdots, L_l\}$ of W' to CSP, where $L_k (1 \le k \le l)$ is denoted as the location of keyword w_k' in keyword set W.

Search($\mathcal{GP}, T, L, I_i, SK_s$) Once receiving the search token T and location set L, CSP first computes $\nu' = e(\pi_0, \alpha)^a$, then he matches the trapdoor with index set I_i to verify whether Eq. 8 holds or not.

$$e(I_0 \cdot \prod_{k=1}^{l} I_{L_k}, T_2) \pi_1^{T_1} = \pi_2 \cdot \nu'^{T_1}. \tag{8}$$

If Eq. 8 holds, then CSP outputs the relevant encrypted file set $C' = \{c_1', \cdots, c_d'\} (1 \le r \le d)$ and the corresponding identity set $ID' = \{id_1', \cdots, id_d'\}$ to PAS. Otherwise, it returns $\perp$.

Verify($\mathcal{GP}, PK_{\mathcal{O}_i}, C', ID'$) After gaining the search results C', PAS checks the accuracy of search results through the following steps:

- **Step 1**: PAS first chooses elements $\tau_r \in_R Z_p^* (1 \le r \le d)$ and sends challenging information $\{r, \tau_r\} (1 \le r \le d)$ to CSP.
- **Step 2**: Then CSP computes the proof information (η, σ) through Eq. 9 and sends it to PAS, where $sig_{i,r} = (H_1(id_r') g_2^{H_2(c_r')})^{b_i}$.

$$\eta = \sum_{r=1}^{d} \tau_r H_2(c_r'), \sigma = \prod_{r=1}^{d} sig_{i,r}^{\tau_r}. \tag{9}$$

- **Step 3**: Finally PAS verifies whether Eq. 10 holds or not.

$$e(\sigma, g_1) = e(\prod_{r=1}^{d} H_1(id_r')^{\tau_r} \cdot g_2^{\eta}, PK_{\mathcal{O}_i}). \tag{10}$$

If Eq. 10 holds, PAS sends C' to DO_i. Otherwise, he aborts this process.

6 Analysis of VMKDO scheme

6.1 Correctness

In this section, we can illustrate the correctness of our scheme if Eq. 8 and 10 hold.

For Eq. 8, we first have $\nu' = e(\pi_0, \alpha)^a = e(g_1^{\lambda}, \alpha)^a = \nu$. And if $W' \subseteq W$ (or $\sum_{k=1}^{l} w_{L_k} = \sum_{k=1}^{l} w'_k$), we can first have

$$\begin{aligned} & e(I_0 \cdot \prod_{k=1}^{l} I_{L_k}, T_2)\pi_1^{T_1} \\ &= e(g_1^{b_i\mu} \cdot g_1^{-\mu\sum_{k=1}^{l} w_{L_k}}, (\beta_i g_1^{-\theta})^{1/(b_i - \sum_{k=1}^{l} w'_k)})\pi_1^{T_1} \\ &= e(g_1^{\mu}, \beta_i)e(g_1^{\mu}, g_1^{-\theta})(\nu \cdot e(g_1, g_1)^{\mu})^{\theta} \\ &= e(g_1^{\mu}, \beta_i) \cdot \nu^{\theta}, \end{aligned}$$

then we get

$$\pi_2 \cdot \nu'^{T_1} = e(g_1, \beta_i)^{\mu} \cdot (e(\pi_0, \alpha)^a)^{T_1} = e(g_1, \beta_i)^{\mu} \cdot \nu^{\theta},$$

finally we verify that Eq. 8 holds.

For Eq. 10, we can first get

$$\begin{aligned} e(\sigma, g_1) &= e(\prod_{r=1}^{d} sig_{i,r}^{\tau_r}, g_1) \\ &= e(\prod_{r=1}^{d} (H_1(id'_r)g_2^{H_2(c'_r)})^{b_i\tau_r}, g_1) \\ &= e(\prod_{r=1}^{d} H_1(id'_r)^{\tau_r} \cdot g_2^{\sum_{r=1}^{d} H_2(c'_r)\tau_r}, g_1^{b_i}) \\ &= e(\prod_{r=1}^{d} H_1(id'_r)^{\tau_r} \cdot g_2^{\eta}, PK_{\mathcal{O}_i}), \end{aligned}$$

then we can check that Eq. 10 holds. Therefore, we prove that our scheme is correct.

6.2 Security

For security, we formally prove that our scheme is secure against KGA in the standard model and can ensure the accuracy of search results. And its security can be guaranteed by the following theorems.

Theorem 1 *Our VMKDO scheme is secure against KGA in the standard model on the condition that the truncated decisional q-ABDHE problem and DBDH problem are intractable.*

Proof In Game 1, our scheme is secure against KGA in the standard model on the condition that the truncated decisional q-ABDHE problem is intractable. As the security proof in Game 1 is similar to the scheme [32, 37], we omit it and just show the detailed security proof in Game 2. Specifically, our scheme can resist KGA assuming that DBDH problem is intractable.

Assume that $\mathcal{A}$ is a polynomial-time adversary which can attack our scheme in Game 2 in the standard model, and $\mathcal{B}$ is a simulator which can play the DBDH game.

Given a tuple $(g_1, g_1^x, g_1^y, g_1^z, Z)$ as the instance for DBDH problem, $\mathcal{A}$'s goal is to decide whether Z is equal to $e(g_1, g_1)^{xyz}$ or to an element in ${}_RG_2$. Then we present the game between $\mathcal{A}$ and $\mathcal{B}$ as follows:

1) **Init**: $\mathcal{B}$ first sets the CSP's public/secret key pair as $PK_s = (A, \alpha), SK_s = x$, where $A = g_1^x, \alpha = g_1^y$, then he selects two elements $\beta_i \in_R G_1, b_i \in_R Z_p^*$ and defines certain DO_i's public/secret key pair as $PK_{\mathcal{O}_i} = (B_i, \beta_i), SK_{\mathcal{O}_i} = b_i$. Finally, he sends the tuple $(PK_s, PK_{\mathcal{O}_i}, SK_{\mathcal{O}_i})$ to $\mathcal{A}$.
2) **Challenge**: $\mathcal{A}$ first submits two target keyword sets (W_0^*, W_1^*), then $\mathcal{B}$ selects a random bit $b \in \{0, 1\}$ and generates the target index $I_{i,b}^*$ for target keyword set W_b^*. $\mathcal{B}$ chooses an element $\mu^* \in_R Z_p^*$ and sets $\nu^* = Z, \pi_0^* = g_1^z, \pi_1^* = Z \cdot e(g_1, g_1)^{\mu^*}, \pi_2^* = e(g_1, \beta_i)^{\mu^*}, I_0^* = B_i^{\mu^*}, I_t^* = g_1^{-w_t\mu^*} (1 \le t \le m)$. Finally, $\mathcal{B}$ returns the tuple $(\pi_0^*, \pi_1^*, \pi_2^*, I_0^*, \{I_t^*\}_{1 \le t \le m})$ to $\mathcal{A}$.
3) **Guess**: $\mathcal{A}$ needs to return a guess bit $b' \in \{0, 1\}$. If $b' = b$, then $\mathcal{B}$ outputs "1" which means that the equation $Z = e(g_1, g_1)^{xyz}$ hold. Otherwise, $\mathcal{B}$ returns "0" which means that Z is an element in ${}_RG_2$.

This completes the proof of Theorem 1. □

Theorem 2 *It is computationally infeasible to generate the valid proof information to pass the result verification for CSP under DL assumption.*

Proof If CSP can pass the result mechanism in a security game [38] through forging a valid proof information on incorrect search results, then we can solve the DL problem in G_1 with an advantage $1 - \frac{1}{p}$. This will contradict to the aforementioned DL assumption as the advantage in breaking the DL problem is negligible. Next, we present the associated security game in detail with the following steps:

– **Step 1**: PAS first sends the challenging information $\{r, \tau_r\}(1 \le r \le d)$ to CSP, and the proof information on correct returned results C' should be (η, σ) such that it can pass the result verification mechanism.

Table 3 Computational complexity in various schemes

Algorithms	Re-dPEKS [32]	Re-dtPECK [33]	VMKDO
KeyGen	$(2\|\mathcal{DO}\|+2)E$	$(\|\mathcal{DO}\|+2)E$	$(\|\mathcal{DO}\|+1)E$
Enc	$(2m+1)E+mP$	$(m+5)E+2P$	$(m+5+2n)E+nH_1+3P$
Trap	$(2l+2)E+H_1$	$(l+3)E$	$2E$
Search	$(l+1)E+H_1+lP$	$(l+4)E+(l+2)P$	$2P+3E$
Verify	—	—	$(d+1)E+dH_1+2P$

1) "$\mathcal{DO}$": Number of authorized DOs; "n": Number of data files;
2) "m": Number of keywords in W; "l": Number of queried keywords;
3) "d": Number of search results; "—": Not having **Verify** algorithm.

- **Step 2**: If CSP returns incorrect search results C^* and forges a proof information (η^*, σ), where $\eta^* = \sum_{r=1}^{d} \tau_r H_2(c_r^*)$, $C' \neq C^*$. Let $\Delta\eta = \eta^* - \eta \neq 0$, if CSP's proof information (η^*, σ) can pass the result verification mechanism, then CSP wins this security game. Otherwise, it fails.
- **Step 3**: Assume that CSP is able to win this game, then we get $e(\sigma, g_1) = e(\prod_{r=1}^{d} H_1(id_r')^{\tau_r} \cdot g_2^{\eta^*}, PK_{\mathcal{O}_i})$ according to Eq. 10. As (η, σ) is correct proof information, we also get $e(\sigma, g_1) = e(\prod_{r=1}^{d} H_1(id_r')^{\tau_r} \cdot g_2^{\eta}, PK_{\mathcal{O}_i})$. Therefore, we reach a conclusion that $g_2^{\eta^* b_i} = g_2^{b_i \eta} \Leftrightarrow g_2^{\Delta\eta b_i} = 1$ according to the properties of bilinear map. However, for two elements $\phi, \varphi \in_R G_1$, there exists an element $\varpi \in_R Z_p^*$ such that $\varphi = \phi^{\varpi}$. Without loss of generality, $g_2^{b_i}$ can be defined as $g_2^{b_i} = \phi^{\rho}\varphi^{\varrho} \in G_1$, where $\rho, \varrho \in_R Z_p^*$. Finally, we have $(\phi^{\rho}\varphi^{\varrho})^{\Delta\eta} = 1 = \phi^{\rho\Delta\eta}\varphi^{\varrho\Delta\eta}$.
- **Step 4**: If CSP wins the game, we can solve the DL problem. Given $\phi, \varphi = \phi^{\varpi} \in_R G_1$, the elements φ, ϖ can be set as $\varphi = \phi^{(-\rho\Delta\eta/\varrho\Delta\eta)}$, $\varpi = -\rho\Delta\eta/\varrho\Delta\eta$ unless $\varrho\Delta\eta = 0$. Whereas, we know that $\Delta\eta \neq 0$ and ϱ is an element in ${}_R Z_p^*$. Thus, the probability of $\varrho\Delta\eta = 0$ is $\frac{1}{p}$ and it is negligible due to the large prime p. Based on above rigorous analysis, we can solve the DL problem with an advantage $1 - \frac{1}{p}$, which will contract to the Definition 2.

Therefore, in our scheme, CSP cannot generate the valid proof information on incorrect search results to pass the result verification under DL assumption, this completes the proof of Theorem 2. □

6.3 Performance

In this section, we mainly assess the performance of our scheme in terms of theoretical performance (computational complexity) and actual performance through exploiting the Type A curves within the Paring Based Cryptography (PBC) library. The experiments are implemented on an Ubuntu 15.04 Server with Intel Core i5 Processor 2.3 GHz using C language and PBC Library. In PBC Library, the Type A is denoted as $E(F_q) : y^2 = x^3 + x$, G_1 is a subgroup of $E(F_q)$, and the cyclic group is a subgroup of $E(F_q)^2$, where q is a large prime number. The group order of G_1 is 160-bit, and the base field is 512-bit.

With respect to theoretical performance, we consider several more time-consuming operations, such as exponentiation operation (E) in G_1 or G_2, pairing operation (P) and hash operation (H_1) which maps a bit string to element in G_1, then we show the computational burden of our scheme through comparing with other analogous schemes [29, 32, 33] in Table 3.

From Table 3, we note that **KeyGen**, **Trap**, **Search** algorithms in our scheme have less computational overhead than those of other schemes. Although **Enc** algorithm in our scheme has heavier computational burden than that of other schemes, it does not affect the user search experience as it just one-time cost. Regarding the particular **Verify** algorithm in our scheme, its computational cost is still acceptable in practice due to the small value of d. Therefore, our scheme is feasible in a broad range of applications as it supports both multi-keyword search and result verification without incurring extra computational burden.

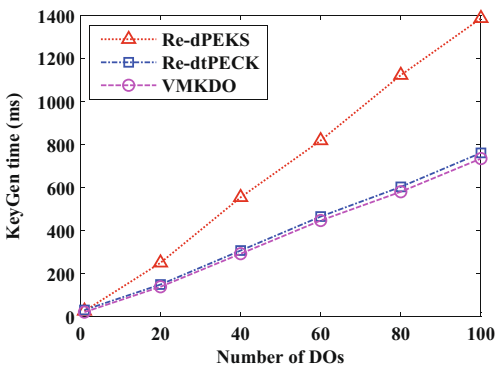

Fig. 4 Key generation time

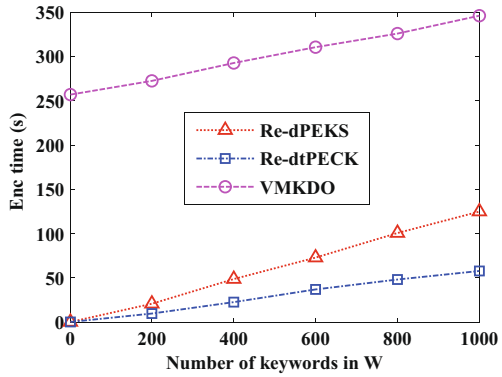

Fig. 5 Ciphertext generation time

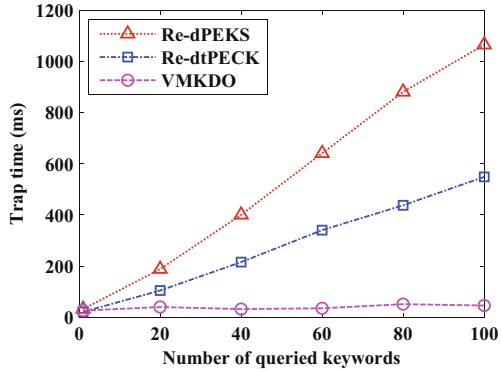

Fig. 6 Trapdoor generation time

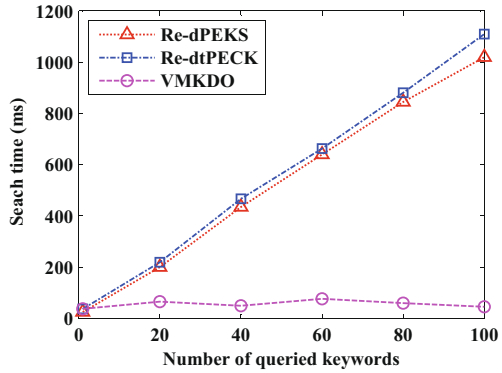

Fig. 7 Ciphertext search time

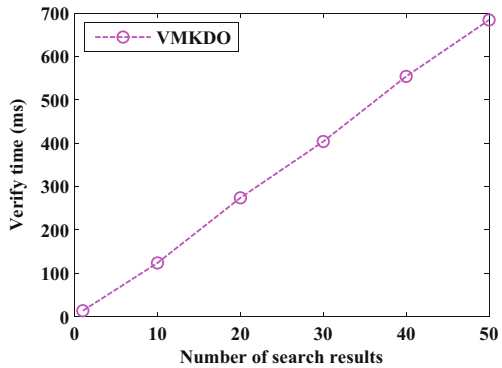

Fig. 8 Result verification time

However, we still need to perform empirical study over a real-world dataset, namely Enron email dataset[1], to evaluate the actual performance of our scheme. For convenience, we first randomly choose 10,000 files ($n = 10000$) from this dataset, set the number of keywords in W be 1000 ($m = 1000$), and then run experiments for 100 times.

In Fig. 4, the computational overhead of **KeyGen** algorithm in all schemes almost linearly increases with the number of DOs (In here, we set $\mathcal{DO} \in [1, 100]$). We notice that our scheme and Re-dtPECK scheme have approximately equal computational overhead in **KeyGen** algorithm, and these two schemes are superior to Re-dPEKS scheme.

As our scheme needs to generate signatures and indexes simultaneously, and both Re-dtPECK and Re-dPEKS schemes just generate indexes, the computational burden of **Enc** algorithm in our scheme is much heavier than that of other two schemes in Fig. 5. For comparison, we set $n = 10000$, thus the **Enc** algorithm in all schemes are just affected by the single factor $m \in [1, 1000]$ and its computational burden becomes heavier with increasing the value of m. Though the performance of **Enc** algorithm in our scheme is inferior to that of other schemes, **Enc** algorithm is just one-time cost and does not affect user search experience. Therefore, our scheme is still acceptable in practice.

From Fig. 6, our scheme has much less computational cost in **Trap** algorithm than other two schemes. And Re-dPEKS scheme is inferior to Re-dtPECK scheme. Besides, the computational costs of **Trap** algorithm in both Re-dtPECK and Re-dPEKS schemes are influenced by the number of queried keywords ($l \in [1, 100]$), and become higher with increasing l, while the computational cost of our scheme almost remains unchanged.

In Fig. 7, both Re-dtPECK scheme and Re-dPEKS scheme in **Search** algorithm have much more computational overhead than our scheme. Moreover, the computational overhead of these schemes is increased with increasing the value of l, while that of our scheme is almost constant.

In Fig. 8, we demonstrate the computational cost of the unique **Verify** algorithm in our scheme. We notice that the computational cost of result verification increases linearly with the number of search results ($d \in [1, 50]$), while it is still within acceptable limits. According to aforementioned comparisons, the actual performance evaluation is in complete accord with theoretical performance shown in Table 3. Therefore, our scheme is efficient and feasible in a broad range of practical applications.

[1] http://www.cs.cmu.edu/~enron/.

7 Conclusion

In this paper, we propose a novel VMKDO scheme to support both result verification and multi-keyword search without incurring heavy computational burden. Besides, our scheme enables DO to issue search queries and delegate his search right to other authorized DO. Different from previous SE schemes, our scheme holds stronger security in resisting keyword guessing attack in the standard model. And empirical experiments over real-world dataset indicate its efficiency and feasibility in practice. As part of future work, we need to explore more efficient SE scheme with supporting expressive search.

Acknowledgments This work was supported by the National High Technology Research and Development Program (863 Program) (No. 2015AA016007, No. 2015AA017203), the Key Program of NSFC (No. U1405255, No. U1135002), the Changjiang Scholars and Innovation Research Team in University (No. IRT1078), the Fundamental Research Funds for the Center Universities (No. JY10000903001) and the Major Nature Science Foundation of China (No. 61370078, No. 61309016).

References

1. Khalil I, Khreishah A, Azeem M (2014) Cloud computing security: a survey. Computers 3(1):1–35
2. Wei LF, Zhu HJ, Cao ZF, Dong XL, Jia WW, Chen YL, Vasilakos A (2014) Security and privacy for storage and computation in cloud computing. Inf Sci 258:371–386
3. Wei LF, Zhu HJ, Cao ZF, Jia WW, Vasilakos A (2010) Seccloud: Bridging Secure Storage and Computation in Cloud. IEEE International Conference on Distributed Computing Systems Workshops. IEEE:52–61
4. Boneh D, Crescenzo GD, Ostrovsky R, Persiano G (2004) Public key encryption with keyword search. International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp 506–522
5. Li HW, Liu DX, Dai YS, Luan TH (2015) Engineering searchable encryption of mobile cloud networks: when QoE meets QoP. IEEE Wirel Commun 22(4):74–80
6. Chai Q, Gong G (2012) Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. IEEE International Conference on Communications. IEEE:917–922
7. Hsien WF, Yang CC, Hwang MS (2016) A survey of public auditing for secure data storage in cloud computing. I J Network Security 18(1):133–142
8. Ren YJ, Shen J, Wang J, Han J, Lee SY (2015) Mutual verifiable provable data auditing in public cloud storage. J Internet Tech 16(2):317–323
9. Song DX, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. IEEE Symposium on Security and Privacy. IEEE:44–55
10. Miao YB, Ma JF, Liu ZQ (2016) Revocable and anonymous searchable encryption in multi-user setting. Concurrency and Computation: Practice and Experience 28(4):1204–1218
11. Xia ZH, Wang XH, Sun XM, Wang Q (2016) A secure and dynamic Multi-Keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352
12. Miao YB, Liu J, Ma JF (2015) Fine-grained searchable encryption over encrypted data in multi-clouds. IEEE International Conference on Wireless Algorithms, Systems, and Applications. IEEE:407–416
13. Fu ZJ, Ren K, Shu JG, Sun XM, Huang FX (2015) Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement. IEEE Transactions on Parallel and Distributed Systems
14. Li J, Wang Q, Wang C, Cao N, Ren K, Lou WJ (2010) Fuzzy keyword search over encrypted data in cloud computing. IEEE International Conference on Computer Communications. IEEE:441–445
15. Wang C, Cao N, Ren K, Lou WJ (2012) Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud data. IEEE Trans Parallel Distrib Syst 23(8):1467–1479
16. Fu ZJ, Sun XM, Liu Q, Zhou L, Shu JG (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans 98-B(1):190–200
17. Boneh D, Waters B (2007) Conjunctive, subset, and range queries on encrypted data. International Conference on Theory of Cryptography. Springer, pp 535–554
18. Hwang YH, Lee PJ (2007) Public key encryption with conjunctive keyword search and its extension to a multi-user system. International Conference on Theory of Cryptography. Springer, pp 2–22
19. Lee CC, Hsu ST, Hwang MS (2013) A study of conjunctive keyword searchable schemes. I J Network Security 15(5):321–330
20. Li HW, Yang Y, Luan TH, Liang XH, Zhou l, Shen XM (2015) Enabling Fine-Grained Multi-keyword Search Supporting Classified Subdictionaries over Encrypted Cloud Data. IEEE Transactions on Dependable and Secure Computing. doi:10.1009/tdsc.2015.2406704
21. Golle P, Staddon J, Waters B (2004) Secure conjunctive keyword search over encrypted data. IEEE International Conference on Applied Cryptography and Network Security. IEEE:31–45
22. Zheng QJ, Xu SH, Ateniese G (2014) VABKS: Verifiable Attribute-based keyword search over outsourced encrypted data. IEEE International Conference on Computer Communications. IEEE:522–530
23. Sun WH, Yu SC, Lou WJ, Hou YT, Li H (2016) Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans Parallel Distrib Syst 27(4):1187–1198
24. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. ACM Conference on Computer and Communications Security. ACM:89–98
25. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. IEEE Symposium on Security and Privacy. IEEE:321–334
26. Waters B (2011) Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. International Conference on Practice and Theory in Public Key Cryptography. Springer, pp 53–70
27. Sun WH, Liu XF, Lou WJ, Hou YT, Li H (2015) Catch you if you lie to me: efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data. IEEE International Conference on Computer Communications. IEEE:2110–2118
28. Miao YB, Ma JF, Wei FS, Liu ZQ, Wang XA, Lu CB (2016) VCSE: Verfiable Conjunctive Keywords Search over Encrypted Data without Secure-channel. Peer-to-Peer Networking and Applications. doi:10.1007/s12083-016-0458-z
29. Shao J, Cao ZF, Liang XH, Lin H (2010) Proxy re-encryption with keyword search. Inf Sci 180(13):2576–2587

30. Fang LM, Susilo W, Ge CP, Wang JD (2012) Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. Theor Comput Sci 462:39–58
31. Wang XA, Huang XY, Yang XY, Liu LF, Wu XG (2012) Further observation on proxy re-encryption with keyword search. J Syst Softw 85(3):643–654
32. Guo LF, Lu B, Li XY, Xu H (2013) A verifiable proxy re-encryption with keyword search without random oracle. IEEE International Conference on Computational Intelligence and Security. IEEE:474–478
33. Yang Y, Ma MD (2016) Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds. IEEE Trans Inf Forensics Secur 11(4):746–759
34. Hu CY, Liu PT (2011) A secure searchable public key encryption scheme with a designated tester against keyword guessing attacks and its extension. IEEE International Conference on Advances in Computer Science, Environment, Ecoinformatics, and Education. IEEE:131–136
35. Rhee HS, Park JH, Lee DH (2012) Generic construction of designated tester public-key encryption with keyword search. Inf Sci 205:93–109
36. Yau WC, Phan RC, Heng SH, Goi BM (2013) Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester. Int J Comput Math 90(12):2581–2587
37. Yang Y (2012) A communication efficient group key distribution scheme for mANETs. IEEE International Conference on Network and System Security. IEEE:361–372
38. Wang BY, Li BC, Li H (2014) Oruta: privacy-preserving public auditingfor shared data in the cloud. IEEE Trans Cloud Computing 2(1):43–56