Received December 30, 2019, accepted February 24, 2020, date of publication March 2, 2020, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977430

A Novel Category Group Index Mechanism for Efficient Ranked Search of Encrypted Cloud Data

LIANGGUI LIU AND QIUXIA CHEN
College of Information Science and Technology, Zhejiang Shuren University, Hangzhou 310015, China

Corresponding author: Lianggui Liu (felix@zjsru.edu.cn)

This work was supported in part by the National Natural Science Foundation of China and Civil Aviation Administration of China under Grant U1533133, in part by the National Natural Science Foundation of China under Grant 61002016 and Grant 61711530653, in part by the Zhejiang Provincial Natural Science Foundation under Grant LQ18F030006, and in part by the China Scholarship Council under Grant 201708330439.

ABSTRACT Nowadays, searchable encryption technology has become the focus of research. Owing to the enormous growth in data storage capacity arising from individual users, business organizations, enterprises and government agencies, it will take these data owners a great deal of time to construct and update indexes in the field of encrypted search in untrusted cloud environment. This has become an urgent problem to be solved. In this paper, based on multiple keywords ranked search, a novel category group index mechanism is proposed. By classifying documents, we create group vectors in each index which can not only transform a high-dimensional secret key into several low-dimensional keys to accelerate the process of encrypting indexes but also improve the flexibility of adding, modifying or deleting documents. When updating indexes, only the group vectors corresponding to the changed category keyword sets need to be updated. During the retrieval process of the proposed mechanism, a novel ''targeted search'' method is designed. With this method, instead of calculating the whole products, cloud server only needs to calculate some inner products of group vectors corresponding to query keywords in the trapdoor and each index to improve the search speed and efficiency. Extensive theoretical analysis and experiment results demonstrate that the method is more feasible and more effective than the compared schemes.

INDEX TERMS Encrypted search, group vector, dimension reduction, category update, targeted search.

I. INTRODUCTION

With the rapid development of computer technology, cloud storage services play a more and more important role in people's daily life. Recently, many individuals and enterprises store their complicated data in the cloud to enjoy some novel high-quality services. By using cloud storage services, they can save local storage costs, share information more efficiently and make use of fast cloud services to process information [1]–[6]. However, many cloud consumers raise concerns about cloud storage security. They worry about that some of their private data will be leaked to the cloud server [7]–[9]. In order to protect the private information of users and enterprises from being leaked, the data need to be encrypted before being stored in the cloud [10]. Although encryption can protect data from attacks by illegal users, unauthorized users and untrusted cloud providers, it will introduce more difficulty to data utilization because encrypted data is more difficult to be searched than plaintext data [11]. In addition, in recent years, the data storage capacity of users and enterprises increases year by year. If all the data are indiscriminately put together for encryption and retrieval, it will not only increase the retrieval complexity but also affects the search efficiency. This problem has become extremely challenging due to the rapidly growing storage needs and untrustworthy cloud storage service. Thus, designing a stable and efficient search mechanism is an important and urgent task.

In order to deal with the fast-growing amount of data, most of the existing encrypted search schemes are complicated and lack of flexibility while updating indexes. Data classification is an effective way to manage massive data storage, while clustering is a common method to solve the problem of data classification, which has become an important means for effective organization and management of text information [12]. When dealing with a large amount of data, documents with similar contents are classified into one category through feature extraction, keyword weight calculation

The associate editor coordinating the review of this manuscript and approving it for publication was Jun Huang.

and clustering [13]. However, existing schemes only focus on classification, and they are not able to provide a secure and efficient cryptographic retrieval mechanism for the classified documents.

Many researchers have made contributions to encrypted search. In order to retrieve encrypted data by entering query keywords, some researchers proposed effective techniques to process encrypted data as a document [14]–[17]. However, the search efficiency of these technologies needs to be improved. Moreover, it is difficult for them to achieve high-quality retrieval for complex data. Thus, these methods cannot be used in complex document retrieval systems directly. Although some methods [18]–[22] were proposed to increase the cryptographic retrieval flexibility, they cannot meet the requirements for complex data search. Meanwhile, they cannot screen out the data that may be useful to the user. Cao *et al.* [23] defined and solved the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). They first proposed a basic idea for the MRSE based on secure inner product computation, and then gave two significantly improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. To improve search experience of the data search service, they further extended these two schemes to support more search semantics. However, their method need high dimensional key to encrypt, and a long time to update indexes for massive data. Therefore, to meet the needs of users, it is important and urgent to design a mechanism that can not only reduce the time of encrypting and updating indexes but also improve the efficiency of retrieval.

In this paper, a highly efficient category group index mechanism (CGIM) was proposed to establish group vectors in the index and to improve the encryption and search speed to overcome the shortcomings of existing methods. Moreover, we take the advantages of document classification to improve the group index mechanism via classifying documents. In the end, the retrieval flexibility can be greatly improved. The contributions of this paper are summarized as follows:

(1) For the first time, we create group vectors in each index by means of classifying documents. The establishment of group vectors can not only transform a high-dimensional secret key into several low-dimensional keys to accelerate the process of encryption but also improve the flexibility of updating indexes without extra cost.

(2) Moreover, during the search process of CGIM, a novel method named targeted search is designed. While searching, with the help of targeted search the cloud server only needs to calculate the inner product of partial group vectors corresponding to the query keywords in each index and search request, which can not only speed up the retrieval process but also increase the searching efficiency.

(3) We provide detailed theoretical analysis including complexity analysis and security analysis of CGIM and MRSE from three aspects: encryption, updating and search to prove that CGIM method is more feasible and more effective than the compared scheme.

The rest of this paper is organized as follows. In section 2, we discuss related work on encrypted search. Section 3 introduces the system model, the threat model, the design goals, the symbols, and the preliminaries. Section 4 describes the proposed CGIM in detail. In Section 5, we detail the extensive theoretical analysis of the complexity and security of CGIM, and compare CGIM with the existing method through extensive experiments. Finally, we conclude this paper in Section 6.

II. RELATED WORK

In 2000, Song *et al.* [24] used untrusted cloud server to search encrypted data remotely and proposed a secure and searchable encryption scheme. This scheme adopted two-layer encryption structure, but the full text search is less efficient. Later, Goh [25] proposed a Z-IDX scheme. In this scheme, the Bloom filter was used as the index structure of a single file, and the keywords contained in the file were mapped into codes and stored in the index. With the help of Bloom filter, data user can determine whether the encrypted file contains a specific keyword. Bellare *et al.* [15] gave as-strong-as-possible privacy definition for public-key encryption schemes, and their RSA-DOAEP construct is the first example of a public-key cipher system. Li *et al.* [16] constructed a fuzzy keyword set with high storage efficiency by editing distance to support privacy protection, and for the first time formally solved the fuzzy search problem. Li *et al.* [22] performed authorized private keyword search over encrypted data, and used the keyword index method to allow multiple users to search. At the same time, a scalable fine-grained authorization framework was proposed, in which users could obtain their search capabilities from local trusted authorities. Curtmola *et al.* [14] proposed a new and stronger security definition of symmetric searchable encryption. Moreover, they also took into account the natural extension, and any group of parties other than the owner can submit search queries. Wong *et al.* [26] discussed the general problems of secure computing on encrypted databases and proposed a SCONEDB model. Then, a new asymmetric scalar product-preserving encryption method was designed and used to construct two secure schemes to support kNN computation on encrypted data.

With the development of searchable encryption technology, more and more intelligent search methods have been proposed. Wang *et al.* [27] defined and solved the problem of secure and effective keyword ranking search over encrypted cloud data, and returned matching files through relevance criteria, which greatly improve the system availability. Therefore, it is closer to the actual situation of private data hosting services in cloud computing. Then they proposed a multi-keyword ranked search scheme for encrypted data [28], which uses a "coordinate matching" similarity method to capture data documents as much as possible, and further use "inner product similarity" to evaluate it. Fu *et al.* [29] studied and solved the problem of personalized multi-keyword ranked search over encrypted data PRSE. Nozoe and Obana [30] proposed a searchable encryption protocol ASSE that supports adding and deleting operations. In ASSE, the index was represented as a simple sequence of bits.

The latest research has made great contributions to the field of searchable encryption. Zhang *et al.* [31] proposed a new dynamic key generation protocol and a data user authentication scheme to effectively authenticate data users and monitor attackers who stole the keys. At the same time, they designed a novel additive order and privacy preserving function family, which implements the ranked search while preserving the privacy-related scores between keywords and files. Wang *et al.* [32] proposed a novel and efficient dynamic multi-keyword ranked search scheme that introduces a reverse data structure to support dynamic operations and use block sparse matrix to improve the search efficiency. Kang and Liu [33] proposed a synonym-searchable encryption scheme for compound bilinear groups, in which data owners rank keywords and their synonyms to improve search efficiency. Then, the cloud transformation server verifies the integrity of results and decrypts the encrypted data, and consequently, the user's burden can be reduced.

Although the above related work can improve the encrypted search efficiency from different aspects, in the environment of big data, the keyword dictionary has a large size that will be related to high encrypted key dimension. Directly using high-dimensional keys to encrypt will lead to high time complexity. Therefore, designing a new method to solve this problem is imminent.

III. PREFERENCE-CONTENT-REGION MODEL

A. SYSTEM MODEL

In cloud computing, the process of encrypted search is performed among three parts, that is, the data owner, the cloud server and the authorized user. The relationship between them is shown in Figure 1. The data owner first uses document clustering method [13] to classify their data before uploading. Then, keywords are extracted from each class, based on which the normalized term frequency is calculated and the indexes are created. Lastly, the encrypted data and indexes are uploaded to the cloud server. While searching, authorized users first create a trapdoor based on the search control, and then submit the trapdoor to the cloud server to query. When the cloud server receives the query trapdoor, it calculates the inner product of the query trapdoor and each document index, and returns results to the authorized user by scores.

According to the description of the system model, there are four stages during the searchable encryption process. **(i)**. Data owner at first creates an index for each document, then use secret key to encrypt the index and document to generate encrypted index set I and encrypted document set C, both of which will be uploaded to cloud server by the data owner. **(ii)**. When the authorized user wants to query data, keyword should be input, then query trapdoor can be generated by search control. Moreover, keyword information cannot be leaked by a trapdoor. **(iii)**. The cloud server use query trapdoor as input to perform search algorithms and return all encrypted documents that include the query keyword. In addition, the cloud server cannot obtain more information despite the returned encrypted documents and the specific search algorithms. **(iv)**. The authorized user decrypts the encrypted data to obtain the query results by access control.

B. DEFINITION OF SECURITY

In the searchable encryption system, the purpose of privacy protection is to prevent the cloud server from inferring any relevant information from the returned results. In this paper, we set the following strict privacy requirements:

Data privacy: data owners use conventional symmetric key encryption technology to encrypt data before it is uploaded to the cloud server, which can successfully prevent the cloud server from spying on outsourced data. If the cloud server can infer the correlation information between keywords and encrypted documents from the index, it can learn the main topics of documents, even the content of documents. Therefore, searchable encrypted indexes should be built to prevent the cloud server from executing this type of association attack. Although the construction of encrypted index can protect the security of an index, the security of all kinds of search information in the process of query is more complex and difficult to solve.

Keyword privacy: due to that users will expose the content of search requests to other people, such as cloud server, the protection of user search information is a very important issue. When a user submits a search request, he wants to protect the keyword information in the search request from being disclosed. Although the trap door generated by encrypting the query request can protect the query keyword information, the cloud server can make some statistical analysis on the search results, so as to disclose the relevant information. For example, as statistics information, the frequency of documents (that is, the number of documents containing keywords) is sufficient for a cloud server to identify keyword content with a high probability [34]. In addition, when the cloud server knows some background information about the dataset, it can use the relevant background information to deduce the key information.

Trapdoor non-correlation: function generated by trapdoor should be random, not deterministic. Particularly, the cloud server should be prevented from inferring the relationship between the trapdoors (for example, whether two trapdoors are generated by the same search request). Otherwise, the cloud server will be able to count the search request frequency of different keywords, which will violate the privacy requirements of keywords. Therefore, to protect the non-correlation of trapdoor is to introduce enough uncertainty into the process of trapdoor generation.

Acquisition mode: during the process of sorting search, access mode returns a sequence of search results, in which each search result is a set of documents with the sequence. For example, the search result of the query key set W is represented as F_W, which includes a sorted list of all document IDs related to W. In order to hide the access mode, some searchable encryption works use privacy information retrieval technology [35].

C. THREAT MODEL

Cloud servers are considered to be "honest but curious" [36], that is, cloud servers will perform search operations honestly, but at the same time, they will infer and analyze encrypted information based on the uploaded data and search trapdoor. These are in line with previous researches on cloud security [37], [38]. In this paper, we assume that the cloud server can obtain the encrypted data, the encrypted index and the trapdoor, and that the cloud server knows how to encrypt the data, but does not know the secret key. Moreover, we assume that the aim of the attack is to try to derive plain information from the encrypted outsourced data. In order to better evaluate the effectiveness of the encryption algorithm, we categorize cloud servers' attacks into two different levels based on how much information is known to the service providers [28].

Level1: The cloud server can know the encrypted data set C, the encrypted index set I, and the query trapdoor T.

Level2: The cloud server know more information than Level1, for example, the cloud server can judge the correlation between query trapdoors by combining existing trapdoors and query results, or use the encrypted background information to infer the encryption keys.

D. DESIGN GOAL

For private users or enterprises with large amount of data, putting documents together for encryption and retrieval will affect the efficiency of creating indexes and the speed of search. If indexes are created according to the classified documents, not only the dimension of the encryption key can be reduced but also the authorized users can generate targeted trapdoor according to the characteristics of query keywords. Due to these facts, in order to consider the security of search process, the following design goals should be achieved.

(1) Creating group vectors in each index: First, we create category keyword sets and a keyword set. Then the group vectors in each index are established based on the category keyword sets.

(2) Targeted search: While searching, only these group vectors in the trapdoor and each index that corresponds to query keywords are involved in the operation, and the group vectors without corresponding to query keywords do not need to be calculated.

(3) Privacy protection: The general goal is to prevent cloud servers from getting private information from the encrypted documents, the encrypted indexes or the search trapdoors. The private information includes keyword information, document content and document category information.

E. SYMBOLIC DESCRIPTION

The notations used in this paper are shown as table 1.

TABLE 1. Notations.

Symbol	Description
F	Plaintext document set, denoted as $F=\{F_1,F_2,\ldots,F_m\}$
C	The set of encrypted set, denoted as $C=\{C_1,C_2,\ldots,C_m\}$
n_j	The total number of keywords in class j
$tf_{ij}^{(x_j)}$	The n_j-th normalized term frequency of class j in the document i
$idf_j^{(x_j)}$	The n_j-th normalized anti-term frequency of class j
W_j	The j-th category keyword set
$\boldsymbol{p}_{ij}$	The j-th group vector of document i
$\boldsymbol{p}_i$	The index of document i
P	The set of plain index
I	The set of encrypted index
$\boldsymbol{q}_j$	The query vector of j-th group
$\boldsymbol{q}$	Query request
Q	Query keyword set
W	Keyword set, $W=\{W_1,W_2,\cdots$
T	Trapdoor

F. PRELIMINARIES

Keyword Set and Category Keyword Set**:** Document clustering method is used to classify the outsourced data and then keywords are extracted from each category to form the category keyword set. All category keyword sets are arranged in sequence to form a keyword set.

Document Score: The importance of each keyword to a document is different. In order to better meet the user's search requirements, we introduce a document score method. The document score is the basis of ranking query results. Term frequency and anti-term frequency are often used to calculate the score of a document. Here, term frequency (TF) indicates the number of times the keyword appears in the document. Anti-term frequency (ATF) represents the number of documents containing the keyword. We use the improved normalized TF·ATF [39] to calculate the score of document F_i when users submit query request $\boldsymbol{q}$. As for keyword set W, we set each dimension of the group vector to the corresponding normalized term frequency when creating indexes, and set each entry of the group vector to the corresponding normalized anti-term frequency when creating a query request.

Top-k query: when users input keywords to query, it is not necessary to return all query results. Only the first few queries with high scores need to be returned. In this paper, we use the

Top-k method to query [36]. When a user submits a query request, he needs to submit parameter k, which indicates the number of results to be returned. During the search process, the cloud server will sort the search results according to the scores, and return the first k results with higher scores to the query users.

Category group index: In this paper, a document index is composed of group vectors. If all documents are divided into k classes, each document index is made up of k group vectors, and each dimension of the group vector is the normalized term frequency of a keyword contained in the corresponding category keyword set.

IV. CATEGORY GROUP INDEX MECHANISM

In this section, we describe our category group index mechanism in detail, which includes six steps.

A. DOCUMENT CLASSIFICATION

While processing data, based on the document classification method proposed in [13], we first extract clustering keywords from every document to identify the same keyword appearing in the document, and then to deal with the same keyword consistently. Secondly, we perform pre-processing to convert the document into a suitable clustering form, including keyword character tagging, removing stop words, and extracting stems. Thirdly, meaning disambiguation is performed to identify the correct meaning of the word and replace the word with its appropriate synonym based on the word context. Fourthly, feature extraction is carried and feature space vectors are created based on these keywords. Finally, we use K-means clustering method [39] to categorize all the documents into k classes according to the feature space vectors. When choosing the parameter k, we try to make the number of documents in each class evenly distributed to optimize the performance.

B. GENERATION OF SECRET KEYS

In this stage, the data owner executes the following algorithm $\{\boldsymbol{S}, \boldsymbol{M}_1, \boldsymbol{M}_2\} = KeyGen(W, \omega)$ to generate secret key, where ω is a given system parameter:

The data owner generates two random invertible matrices $\boldsymbol{M}_1, \boldsymbol{M}_2$ and the indicator $\boldsymbol{S}$ by

$$\boldsymbol{M}_1 = \begin{pmatrix} \boldsymbol{M}_{11} & 0 & \cdots & 0 \\ 0 & \boldsymbol{M}_{12} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \boldsymbol{M}_{1k} \end{pmatrix},$$

$$\boldsymbol{M}_2 = \begin{pmatrix} \boldsymbol{M}_{21} & 0 & \cdots & 0 \\ 0 & \boldsymbol{M}_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \boldsymbol{M}_{2k} \end{pmatrix}, \quad \text{and}$$

$$\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_k).$$

Here, $\boldsymbol{M}_{1j}$ and $\boldsymbol{M}_{2j}(j \;=1,2,\ldots,\; k)$ are $(n_j + u + 1) \times (n_j + u + 1)$ random invertible block matrices, and $\boldsymbol{s}_j \in \{0,1\}^{(n_j+u+1)}(j \;=1,2,\ldots,\; k)$, where n_j is the total number of keywords of the j-th category and u is the extended dimension that equals 2ω. The ultimate secret key can be described as a three tuple $\{\boldsymbol{S}, \boldsymbol{M}_1, \boldsymbol{M}_2\}$.

C. CREATION OF GROUP VECTORS IN EACH INDEX

The index generation algorithm I = Encrypt(F, W, $\{\boldsymbol{S}, \boldsymbol{M}_1, \boldsymbol{M}_2\}$) is described in Algorithm 1.

Algorithm 1 The Index Generation Algorithm Pseudocode

```
Input: the document set $F = \{F_1, F_2, \ldots, F_m\}$, the
       keyword set $W = \{W_1, W_2, \ldots, W_k\}$ and the secret
       key $\{M_1, M_2, S\}$
Output: the encrypted index $I$.
1: for $i = 1$ to $m$:
2:     for $j = 1$ to $k$:
3:             Create an $n_j$ dimensional vector $p_{ij}$,
               where $n_j$ is the size of $W_j$.
4:         for $a = 1$ to $n_j$:
5:                 Set $p_{ij}[a]$ to the normalized term
                   frequency of $w_{ja}$ in $F_i$, where $w_{ja}$
                   is the $a$-th keyword in $W_j$.
6:         end for
7:         Extend $p_{ij}$ from $n$ dimension to $n+u+1$
           dimensionand set $p_{ij}[l]$ ($l \in [n+1, n+u+1]$)
           to a random number $\rho^{(l)}$.
         Randomly split $p_{ij}$ into two vectors $p'_{ij}$ and $p''_{ij}$.
8:         for $b = 1$ to $n+u+1$:
9:             if $s_j[b] = 0$:
10:                $p_{ij}[b] = p'_{ij}[b] = p''_{ij}[b]$;
11:            else
12:                $p_{ij}[b] = p'_{ij}[b] + p''_{ij}[b]$;//($p'_{ij}[b] \neq p''_{ij}[b]$,
                   $p'_{ij}[b] \neq 0$ and $p''_{ij}[b] \neq 0$)
13:            end if
14:        end for
15:    end for
16:    $p'_i = (p'^T_{i1}, p'^T_{i2}, \ldots, p'^T_{ik})^T, p''_i$
           $= (p''^T_{i1}, p''^T_{i2}, \ldots, p''^T_{ik})^T$.
17:    $I_i = \{M_1^T p'_i, M_2^T p''_i\}$.
18: end for
19: $I = \{I_1, I_2, \ldots I_m\}$.
```

D. CREATION OF TRAPDOOR

After the user input the keyword for inquiry, the trapdoor generation algorithm T = Encrypt(Q, W, $\{\boldsymbol{S}, \boldsymbol{M}_1, \boldsymbol{M}_2\}$) should be performed, which is described as Algorithm 2.

E. TARGETED SEARCH

The search algorithm C = Search(I, T, k) is described as follows: The authorized user uploads the query trapdoor to the cloud server. When the cloud server receives the trapdoor, it first finds out these group vectors that are not equal to $\vec{0}$ in the trapdoor. Then the secure inner product of these group vectors and the corresponding group vectors in each

Algorithm 2 The Trapdoor Generation Algorithm Pseudocode

```
Input: the query keyword set $Q$, the keyword set
        $W = \{W_1, W_2, \dots, W_k\}$ and the secret
        key $\{\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{S}\}$
Output: the trapdoor $T$
1: for $j = 1$ to $k$:
2:      Create an $n_j$ dimensional vector $\boldsymbol{q}_j$,where $n_j$ is the
        size of $W_j$
3:      for $a = 1$ to $n_j$:
4:          if $Q$ contains $w_{ja}$: // $w_{ja}$ is the $a$-th keyword
            in $W_j$.
5:              Set $\boldsymbol{q}_j[i]$ to the anti-term frequency of $w_{ja}$.
6:          else
7:              Set $\boldsymbol{q}_j[i]$ to 0.
8:          end if
9:      end for
10:     Extend $\boldsymbol{q}_j$ from $n$ dimension to $n + u + 1$ dimension.
        For the locations from $(n+1)$ to $(n+u)$, by randomly
        choosing $v$ out of $u$ positions, the corresponding
        entries are set as 1 and the remaining entries are set as
        0.
        The value of $(n+u+1)$ dimension is set as a random
        number $t(t \neq 0)$ and all the other locations are
        multiplied by another random number $r(r \neq 0)$.
        Randomly split $\boldsymbol{q}_j$ into two vectors $\boldsymbol{q}'_j$ and $\boldsymbol{q}''_j$.
11:     for $b = 1$ to $n + u + 1$:
12:         if $s_j[b] = 0$:
13:             $\boldsymbol{q}_j[b] = \boldsymbol{q}'_j[b] = \boldsymbol{q}''_j[b]$;
14:         else
15:             $\boldsymbol{q}_j[b] = \boldsymbol{q}'_j[b] + \boldsymbol{q}''_j[b]$; //($\boldsymbol{q}'_j[b]$
                $\neq \boldsymbol{q}''_j[b], \boldsymbol{q}'_j[b] \neq 0$ and $\boldsymbol{q}''_j[b] \neq 0$)
16:         end if
17:     end for
18: end for
19: $\boldsymbol{q}' = (\boldsymbol{q}'_1, \boldsymbol{q}'_2, \dots, \boldsymbol{q}'_k), \boldsymbol{q}'' = (\boldsymbol{q}''_1, \boldsymbol{q}''_2, \dots, \boldsymbol{q}''_k)$.
20: $T = \{\boldsymbol{M}_1^{-1}\boldsymbol{q}', \boldsymbol{M}_2^{-1}\boldsymbol{q}''\}$.
```

document index are calculated. Finally, the top k encrypted documents with the highest score are returned to the authorized user. For example, the documents we want to query are in class j, the search process is as follows:

$$\begin{aligned} I_i \cdot T &= \{\boldsymbol{p}'_{i\sim}\boldsymbol{M}_1, \boldsymbol{p}''_{i\sim}\boldsymbol{M}_2\} \cdot \{\boldsymbol{M}_1^{-1}\boldsymbol{q}'^{T}_{\sim}, \boldsymbol{M}_2^{-1}\boldsymbol{q}''^{T}_{\sim}\} \\ &= \boldsymbol{p}'^{T}_{ij\sim}\boldsymbol{q}'_{j\sim} + \boldsymbol{p}''^{T}_{ij\sim}\boldsymbol{q}''_{j\sim} \\ &= \boldsymbol{p}^{T}_{ij\sim}\boldsymbol{q}_{j\sim} \\ &= r(\boldsymbol{p}^{T}_{ij}\boldsymbol{q}_j + \sum \rho_i^{(v)}) + t \\ &= r(\text{Score}(F_i, \boldsymbol{q}) + \sum \rho_i^{(v)}) + t \end{aligned}$$

where ρ is a random number that will cause the leakage of information in Level2 attack model [40].

After receiving the encrypted document C returned by the cloud server, the authorized user first obtains the decryption key sk through the access control operation, and then uses decryption algorithms $F' = \text{Dec}(sk, C')$ to decrypt the encrypted document C' returned by cloud server into a plaintext document F'.

F. UPDATING INDEX

The updating operations consist of three parts, adding new documents, modifying existing documents, and deleting existing documents, each of which will affect the length of existing indexes. The index dimension of MRSE scheme is fixed and the same as the total number of keywords in the original keyword dictionary. Thus, all indexes need to be re-encrypted when an updating operation is performed, which will result in a huge amount of computation. Subsequently, the scheme has been improved to reduce such great cost of updating by reserving some blank entries in the dictionary and set corresponding positions in each index as zero. However, this method is not suitable for practical situation because we can't predict the user's demand. In our proposed mechanism, we classify the documents to extract keywords from each class to construct category keyword sets. Then, based on these category keyword sets, group vectors are established in each index to improve the flexibility of updating indexes. If users update documents, they first determine which class need to be updated, then update the group vectors corresponding to these classes in each index, and unrelated group vectors remain unchanged. For example, to update the document in the j class, the update process is as follows:

Step 1 : We re-extract keywords from the j-th updated documents to construct the j-th category keyword set, and then regenerate the j-th block matrices $\boldsymbol{M}_{j1}, \boldsymbol{M}_{j2}$ and the indicator $\boldsymbol{s}_j$ according to the total number of updated keywords.

Step 2: We need to re-create the group vector corresponding to the j-th category keyword set in each index, and the length of the group vector is the same as the total number of updated keywords. Rules of creating group vectors are as follows: If document F_i belongs to the j-th category, the normalized term frequency of each dimension in the j-th group vector is recalculated. If document F_i is not in class j, each dimension of the corresponding group vector is set to 0. Then, the j-th group vector is extended, split and encrypted to generate an encrypted group index according to the procedures mentioned above. Particularly, when adding a new document, we need to create the other group vectors to generate the document index. Moreover, when a document is deleted, the corresponding index also needs to be removed.

Step 3: We replace the j-th group vector in each index, and then upload the updated indexes to the cloud server.

It should be noted that updating documents will affect the IDF value of the keyword in existing dictionary. The data owner needs to recalculate the IDF values of all keywords when adding, modifying, or deleting documents, and send them to authorized users. Xia *et al.* [41] have proved that when the number of documents added or deleted by the data owner is less than 300, the IDF value of the keyword will not change much, that is, the error is negligible. At this time, the data owner does not need to recalculate all keyword IDF values. When the update operation causes the keyword IDF value to change greatly, the data owner can flexibly select those keywords whose IDF value changes greatly to update.

V. PERFORMANCE ANALYSIS

A. THEORETICAL ANALYSIS

1) COMPLEXITY ANALYSIS

Below we compare the complexity of MRSE and CGIM from three aspects: encryption, updating and search. Analyses are given in detail:

In terms of encryption complexity, MRSE uses $(N + U + 1) \times (N + U + 1)$ dimensional invertible matrix to encrypt $(N + U + 1)$ dimensional indexes and query requests. The complexity of encrypting m document indexes is approximately $O(m(N + U)^2)$. Where N represents the total number of keywords, satisfying $N \approx \sum_{j=1}^{k} n_j$, U represents the extended dimension, satisfying $U = \sum_{j=1}^{k} u$. Therefore, the encryption complexity can be further described as $O(m(\sum_{j=1}^{k} (n_j + u))^2)$. However, the index and query request in CGIM are composed of k group vectors, and the group vector dimension is $(n_j + u + 1)(j = 1, 2, \ldots, k)$. The complexity of encrypting m document indexes is $O(m \sum_{j=1}^{k} (n_j + u)^2)$. When k >1, $\sum_{j=1}^{k} (n_j + u)^2 < (\sum_{j=1}^{k} (n_j + u))^2$, so the encryption complexity of CGIM is less than MRSE. In addition, according to Cauchy inequality, we can infer that $\sum_{j=1}^{k} (n_j + u)^2 \geq (\sum_{j=1}^{k} (n_j + u))^2/k$, so when the number of keywords in each group is equal, the encryption complexity of CGIM will approach the minimum value $O(m(N + U)^2/\mathrm{k})$.

In terms of updating complexity, the MRSE needs to re-encrypt the index. Therefore, the updating complexity is $O(m(N + U)^2)$. CGIM only need to update parts of the group vector. When updating $c(c \leq k)$ classes of documents, the encryption complexity is $O(m \sum_{c} (n_j + u)^2)$. Therefore, the updating complexity of CGIM is less than MRSE, and when the number of keywords in each class is equal, the average updating complexity will approach a minimum value $O(mc(N + U)^2/k^2)$.

In terms of search complexity, MRSE calculates the inner product of query trapdoor and each document index during the search process. The document index and query request are both $(N + U + 1)$ dimensional vectors, so the search complexity is $O(M(N + U))$, which can also be expressed as $O(m \sum_{j=1}^{k} (n_j + u))$. However, CGIM only needs to calculate the inner product of partial group vectors. When querying $c(c \leq k)$ classes of documents, the query complexity is $O(m \sum_{c} (n_j + u))$. Therefore, the search complexity of CGIM is not greater than MRSE. When the number of keywords in each class is equal, the average search complexity will have a minimum value $O(mc(N + U)/k)$.

2) SECURITY ANALYSIS

Protecting the security of information is a critical step in searchable encryption process. In Level2 attack model, we will investigate the security of our proposed method in terms of the key information, the keyword information, the query information and the trapdoor non-correlation.

The secret key security: In Level2 attack model, the cloud server knows the index encryption process $I_i = \{\boldsymbol{p}'_{i\sim}\boldsymbol{M}_1, \boldsymbol{p}''_{i\sim}\boldsymbol{M}_2\}$.To enhance the key protection, we further assume that the cloud server knows the size of each group vector and knows the encryption process. In other words, the encryption process of the j-th (j = 1,2,…, k) group vector is $\{\boldsymbol{p}'^T_{ij\sim}\boldsymbol{M}_{j1}, \boldsymbol{p}''^T_{ij\sim}\boldsymbol{M}_{j2}\}$. However, the cloud server does not know the specific steps of the segmentation. We denote the j-th encrypted group index as $\widehat{\boldsymbol{p}}_{ij1}$ and $\widehat{\boldsymbol{p}}_{ij2}$, and the total number of documents is m. For $\boldsymbol{p}'_{ij\sim}$ and $\boldsymbol{p}''_{ij\sim}$, it can only be set as an $n_j + u + 1$ dimensional random vector. Then the following equation can be established:

$$\begin{cases} \boldsymbol{p}'^T_{ij\sim}\boldsymbol{M}_{j1} = \widehat{\boldsymbol{p}}_{ij1} \\ \boldsymbol{p}''^T_{ij\sim}\boldsymbol{M}_{j2} = \widehat{\boldsymbol{p}}_{ij2} \end{cases}$$

where $\boldsymbol{p}'_{ij\sim}$ and $\boldsymbol{p}''_{ij\sim}$ have $2(n_j + u + 1)m$ unknown variables, $\boldsymbol{M}_{j1}$ and $\boldsymbol{M}_{j2}$ have $2(n_j + u + 1)^2$ unknown variables, and now there are only $2(n_j + u + 1)m$ equations. Thus, the cloud server does not have enough information to infer the encryption matrix.

The keyword security: In Level2 attack model, the leakage of information is mainly caused by the random number ρ [40]. In order to meet the security requirement which is defined beforehand, when the system parameters ω are given, we require that the identical probability of two $\sum \rho^{(v)}$ is less than $1/2^{\omega}$. That is, there should be at least 2^{ω} different values of $\sum \rho^{(v)}$. On the other hand, $\sum \rho^{(v)}$ has C_u^v different values, and when $u/v = 2$, C_u^v will be the maximum value. Therefore, when $u = 2\omega$ and $v = \omega$, the scheme can achieve the security goal. In addition, each $\rho^{(j)}$ follows the same uniform distribution $U(\mu' - c, \mu' + c)$, and their mean is μ' and variance is $\sigma' = c^2/3$. According to the central limit theorem, $\sum \rho^{(v)}$ follows normal distribution whose mean will be $\omega\mu'$, and variance is $\omega\sigma' = \omega c^2/3$. Let $\sum \rho^{(v)}$ follow the standard normal distribution $N(\mu, \sigma^2)$, and set μ' to μ/ω and c to $\sqrt{3/\omega}\sigma$. In the standard normal distribution, the larger the value of σ, the harder it is for the cloud server to infer the similarity information between initial index vectors, but the search accuracy will be reduced. Therefore, it is necessary to choose the value of σ reasonably between the privacy protection and the search accuracy.

The security of the query information and the trapdoor non-correlation: In order to prevent the cloud server from inferring

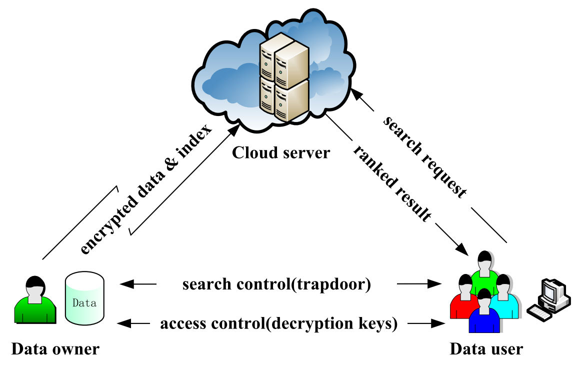

FIGURE 1. Searchable encryption system model.

users' query information from trapdoors, it is required to hide the users' query information when submitting trapdoors. CGIM performs dimensional extension, random split and encryption on the group vectors containing query keywords. After encryption, the query keyword information will not directly appear in the query trapdoor. Thus, it is safe for query information. Furthermore, for different query requests, the introduction of random number r and t makes a document have different score even if users enter the same query keywords. Therefore, it can protect the trapdoor non-correlation.

B. EXPERIMENTAL ANALYSIS

We have done extensive experiments to evaluate the performance of CGIM. As for the experimental environment, a searchable encryption system was built on Windows 7 server and experimental verification were carried out, where the CPU is Intel core i5 (2.5GHz) processor. Moreover, we use RFC (Request For Comments) [42] as the data set, and java language to implement the experimental system.

The experimental process is described as follows. First, we use document clustering method to divide documents into two and four classes. Then, for MRSE and CGIM, three keywords are extracted from each document to form the keyword set and the category keyword set, and the total number of keywords is 15630. Given the system parameter $\omega = 2000$, in MRSE, each index is extended by $2\omega + 1$ dimensions. In CGIM, each group vector of the index is extended by $2\omega/k+1$ dimensions, where k is the number of groups. Meanwhile, during the process of trapdoor creation in both of the two methods, we set the query keywords to 10. We compare our method with MRSE to verify the better performance of our proposed method in the fields of index generation, creation of trapdoor, updating index and targeted search. When the documents are divided into two or four classes and there is significant difference in the numbers of keywords in each group, our methods are named CGIM2 and CGIM4, respectively. Moreover, we use CGIM2(1) and CGIM4(1) to denote the methods when the documents are divided into two or four classes and the numbers of keywords in each group exhibit approximately uniform distribution.

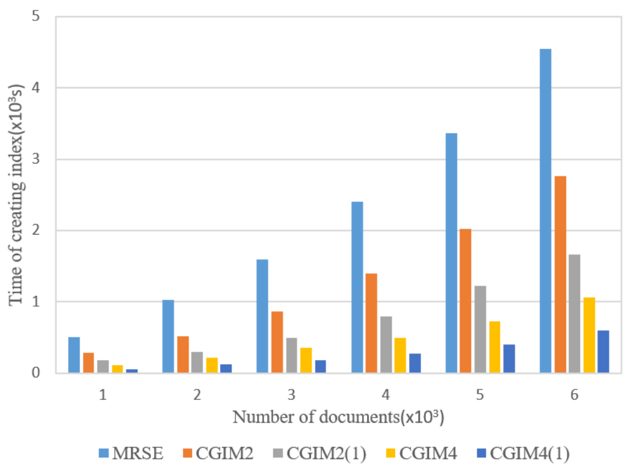

FIGURE 2. Time of creating index. For different number of documents with the system parameter $\omega = 2000$.

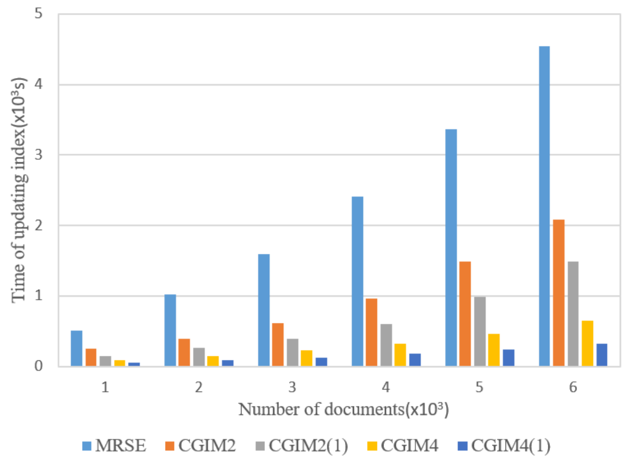

FIGURE 3. Time of updating index. For different number of documents with the number of updating class $c = 1$.

The process of generating an index consists of creating and encrypting each group vector in the index. The relationship between the time of creating indexes and the number of documents is shown in Fig. 2. From Fig. 2 we can see that for the same number of documents, time of creating a category group index is less than that of MRSE scheme, and for the same number of documents, the time needed grew in inverse proportion to the number of classes. When a user wants to add, modify or delete a document, all indexes need to be updated.

Fig. 3 shows the time of updating indexes for MRSE and CGIM. On the one hand, the larger the number of documents, the larger the size of the keyword dictionary, which will result in the higher the corresponding index dimension and encryption matrix dimension. In the end, the calculation of the product of the two will increase dramatically. Therefore, the time consumption of the two schemes increases with the number of documents. On the other hand, for the same number of documents, the length of keyword dictionary of MRSE

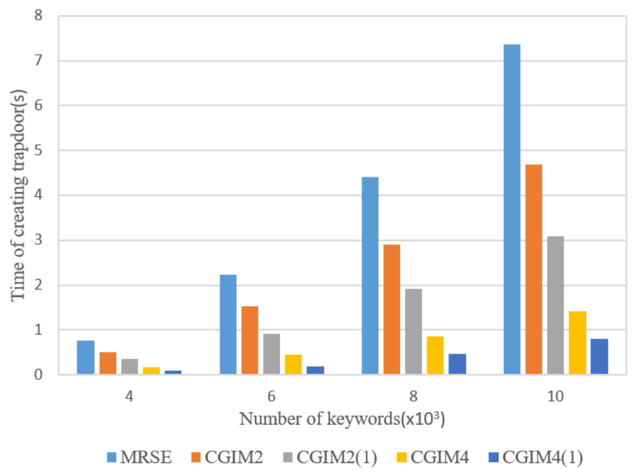

FIGURE 4. Time of creating trapdoor. For different number of documents with the same query keywords $t = 10$.

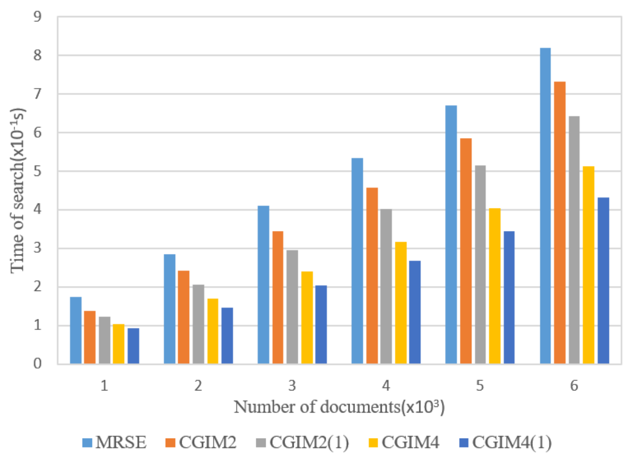

FIGURE 5. Time of search. For different number of documents with the same query keywords $t = 10$.

and CGIM is approximately the same. As can be seen from the figure, the time of updating indexes CGIM is less than that of MRSE scheme. In both Fig. 2 and Fig. 3, the encryption time needed grew in inverse proportion to the number of classes, as shown in CGIM2(1) and CGIM4(1). When the number of keywords in each group increases, the encryption time will increase based on the original trend, as shown in CGIM2 and CGIM4.

The relationship between the update encryption time and the number of documents is shown in Fig. 4. From the figure, we can see that the more documents, the larger the keyword dictionary size and the higher the index and key dimension there are, and in the end the longer it takes to update the encryption. Moreover, we can see when updating the same documents, the average encryption time of CGIM is less than that of MRSE. When the distribution difference of key words in each group increases, the average update encryption time will also increase based on the original trend, as shown in CGIM 2 and CGIM 4 in the figure. In addition, the more evenly distributed each group of key words, the smaller the average update encryption time will be; and the more groups, the less the update encryption time will be, as shown in CGIM2(1) and CGIM4(1).

Fig. 5 shows the relationship between the time of search and the number of documents. From the figure, we can see the time of search increases with the number of documents. The main reason is that both schemes need to calculate the vector inner product of the query trapdoor and each document index, and the more documents there are, the more computation will occur, so the longer the time it will take. CGIM uses targeted search method to save the calculation of unrelated group vectors. When we input the same keyword query, the average query time of CGIM will not be longer than that of MRSE. When the distribution difference of key words in each group increases, the average query time will increase on the basis of the original trend, as shown in the figure CGIM2 and CGIM4. In addition, the more uniform the number of key words in each group is, the less the average query time is; and it decreases with the increase of the number of groups, as shown in CGIM2(1) and CGIM4(1).

VI. CONCLUSION

In this paper, we proposed category group index mechanism for efficient ranked search of encrypted cloud data. Unlike existing encrypted search schemes, we classify the documents and create group vectors in each index to reduce the dimension of encryption keys. And in the end, the time of creating index is saved. At the same time, the establishment of group vectors in each index also facilitates users to update their data. When users want to add, modify or delete documents, they only need to recreate the group vectors corresponding to the updated category keyword sets. Besides, in the retrieval process of CGIM, we propose a ‘‘targeted search’’ method. With this method, the cloud server only needs to calculate the products of some group vectors corresponding to query keywords in the trapdoor and each index, instead of the whole products of them, thus improving the speed and efficiency of search. From the theoretical and experimental analysis, our method shows better performance than its counterparts.

REFERENCES

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, ‘‘A break in the clouds: Towards a cloud definition,’’ *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2008.
[2] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, ‘‘LT codes-based secure and reliable cloud storage service,’’ in *Proc. Proc. IEEE INFOCOM*, Mar. 2012, pp. 693–701.
[3] J. Lee, ‘‘A view of cloud computing,’’ *Int. J. Networked Distrib. Comput.*, vol. 1, no. 1, p. 2, 2013.
[4] Y. Miao, J. Weng, X. Liu, K.-K. Raymond Choo, Z. Liu, and H. Li, ‘‘Enabling verifiable multiple keywords search over encrypted cloud data,’’ *Inf. Sci.*, vol. 465, pp. 21–37, Oct. 2018.
[5] L. Zhang, Y. Zhang, and H. Ma, ‘‘Privacy-preserving and dynamic multi-attribute conjunctive keyword search over encrypted cloud data,’’ *IEEE Access*, vol. 6, pp. 34214–34225, 2018.
[6] H. Lian, W. Qiu, D. Yan, Z. Huang, and J. Guo, ‘‘Efficient privacy-preserving protocol for k-NN search over encrypted data in location-based service,’’ *Complexity*, vol. 2017, Dec. 2017, Art. no. 1490283.

[7] F. Han, J. Qin, and J. Hu, "Secure searches in the cloud: A survey," *Future Gener. Comput. Syst.*, vol. 62, pp. 66–75, Sep. 2016.
[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," *Future Gener. Comput. Syst.*, vol. 30, pp. 179–190, Jan. 2014.
[9] X. Ge, J. Yu, C. Hu, H. Zhang, and R. Hao, "Enabling efficient verifiable fuzzy keyword search over encrypted data in cloud computing," *IEEE Access*, vol. 6, pp. 45725–45739, 2018.
[10] Z. Mei, H. Zhu, Z. Cui, Z. Wu, G. Peng, B. Wu, and C. Zhang, "Executing multi-dimensional range query efficiently and flexibly over outsourced ciphertexts in the cloud," *Inf. Sci.*, vol. 432, pp. 79–96, Mar. 2018.
[11] J. Li, J. Li, X. Chen, Z. Liu, and C. Jia, "Privacy-preserving data utilization in hybrid clouds," *Future Gener. Comput. Syst.*, vol. 30, pp. 98–106, Jan. 2014.
[12] D. Sánchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 7718–7728, Jul. 2012.
[13] L. Ge and T.-S. Moh, "Improving text classification with word embedding," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2017, pp. 1796–1805.
[14] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *J. Comput. Secur.*, vol. 19, no. 5, pp. 895–934, 2011.
[15] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2007, pp. 535–552.
[16] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.
[17] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith, "Public key encryption that allows PIR queries," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2007, pp. 50–67.
[18] Z. Liu, T. Li, P. Li, C. Jia, and J. Li, "Verifiable searchable encryption with aggregate keys for data sharing system," *Future Gener. Comput. Syst.*, vol. 78, pp. 778–788, Jan. 2018.
[19] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proc. Int. Conf. Pairing-Based Cryptogr.* Berlin, Germany: Springer, 2007, pp. 2–22.
[20] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," *J. Cryptol.*, vol. 26, no. 2, pp. 191–224, Feb. 2012.
[21] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2010, pp. 62–91.
[22] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 383–392.
[23] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, Jan. 2014.
[24] D. Xiaoding Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. Proc. IEEE Symp. Secur. Privacy. S&P*, May 2000, pp. 44–55.
[25] E.-J. Goh, "Secure indexes," in *Proc. IACR*, Oct. 2003, p. 216.
[26] W. K. Wong, D. W.-L. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. 35th SIGMOD Int. Conf. Manage. Data*, 2009, pp. 139–152.
[27] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst.*, Jun. 2010, pp. 253–262.
[28] H. Yin, Z. Qin, J. Zhang, L. Ou, F. Li, and K. Li, "Secure conjunctive multi-keyword ranked search over encrypted cloud data for multiple data owners," *Future Gener. Comput. Syst.*, vol. 100, pp. 689–700, Nov. 2019.
[29] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.
[30] S. Nozoe and S. Obana, "Searchable symmetric encryption supporting update," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, Nov. 2016, pp. 713–717.
[31] 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, May 2016.
[32] J. Wang, X. Chen, H. Ma, Q. Tang, J. Li, and H. Zhu, "A verifiable fuzzy keyword search scheme over encrypted data," *J. Internet Serv. Inf. Secur.*, vol. 2, nos. 1–2, pp. 49–58, 2012.
[33] Y. Kang and Z. Liu, "A fully secure verifiable and outsourced decryption ranked searchable encryption scheme supporting synonym Query," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2017, pp. 223–231.
[34] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+R: Top-k retrieval from a confidential index," in *Proc. 12th Int. Conf. Extending Database Technol. Adv. Database Technol. (EDBT)*, Saint Petersburg, Russia, 2009, pp. 24–26.
[35] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Cryptography from anonymity," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2006, pp. 239–248.
[36] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 917–922.
[37] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[38] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[39] M. P. Naik, H. B. Prajapati, and V. K. Dabhi, "A survey on semantic document clustering," in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Mar. 2015, pp. 1–10.
[40] Z. Fu, X. Wu, Q. Wang, and K. Ren, "Enabling central keyword-based semantic extension search over encrypted outsourced data," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 2986–2997, Dec. 2017.
[41] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 340–352, Feb. 2016.
[42] Internet Society. (Nov. 18, 2019). *Request for Comments*. [Online]. Available: http://www.ietf.org/rfc.html

LIANGGUI LIU received the Ph.D. degree in communications and information system from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2007. He was a Visiting Professor with Cornell University, Ithaca, NY, USA, and the University of Houston, Houston, TX, USA. His current research interests focus on social computing, network security, and natural computation. He is currently a Professor with the College of Information Science and Technology, Zhejiang Shuren University, Hangzhou, China.

QIUXIA CHEN received the Ph.D. degree from the Department of Automation, Zhejiang University of Technology, China, in 2011. She is currently an Associate Professor with the College of Information Science and Technology, Zhejiang Shuren University, Hangzhou, China. Her current research interests focus on model predictive control and network security.