International Conference on Advances in Materials, Computing and Communication Technologies
In Association withInternational Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

An Efficient Semantic Aware Search Method over Encrypted Cloud Data

Mrs.Viswanathan Manimala, Mr.D.Mohan

[1]Department of Computer Science and Engineering, Global Institute Of Engineering and Technology, Tamil Nadu, India

[2]Assistant Professor, Department of Computer Science and Engineering, Global Institute Of Engineering and Technology, Tamil Nadu, India

ABSTRACT

With the increasing adoption of cloud computing, a growing number of users outsource their to cloud. To preserve the privacy, they are usually encrypted before outsourcing. However, the common practice of encryption makes the effective utilization of the data difficult. For example, it is difficult to search the given keywords in encrypted. Many schemes are proposed to make encrypted data searchable based on keywords. However, keyword-based search schemes ignore the semantic representation information of user's retrieval, and cannot completely meet with users search intention. Therefore, how to design a content-based search scheme and make semantic search more effective and context-aware is a difficult challenge. In this paper, we propose ECSED, a novel semantic search scheme based on the concept hierarchy and the semantic relationship between concepts in the encrypted. ECSED uses two cloud servers. One is used to store the outsourced and return the ranked results to data users. The other one is used to compute the similarity scores between the documents and the query and send the scores to the first server. To further improve the search efficiency, we utilize a tree-based index structure to organize all the document index vectors. We employ the ranked search over encrypted cloud data as our basic frame to propose two secure schemes. The experiment results based on the real world show that the scheme is more efficient than previous schemes. We also prove that our schemes are secure under the known model and the known background model.

Index Terms -Public cloud, Results Vertifiable searching, secure semantic searching , word transporation.

I. INTRODUCTION

Cloud computing refers to accessing software and storing data in the cloud of the internet. It is a model for enabling convenient, on demand network access to a shared pool of configurable and reliable computing resources that can be rapidly provisioned and released with service provider interaction. The security of outsourced data cannot be guaranteed, as the Cloud Service Provider (CSP) possesses whole control of the data. So, it is necessary to encrypt data before outsourcing them into cloud to protect the privacy of sensitive data. The idea of proposed system comes from many researchers have proposed a series of efficient search schemes over encryptedclouddata.Alltheexistingsearchableencrypti onschemes,which consider keywords as the document feature, do not take the semantic relations between words into consideration. The semantic relations between words are diverse [8], such as synonymy and domain correlation. Considering the potentially huge amount of outsourced data documents in the cloud, the search accuracy and search efficiency are influenced negatively if the semantic relations between words are not handled well. In this paper, proposes an efficient searchable encrypted scheme based on concept hierarchy supporting semantic search with two cloud servers. A concept hierarchy tree is constructed based on domain concepts related knowledge of the outsourced dataset. The concept hierarchy is extended to include more semantic relations between concepts. With the help of extended concept hierarchy, document features are extracted more precisely and search terms are well extended based on the semantic relations between concepts.

II. LITERATURE

In recent years, many researchers have proposed a series of efficient search schemes over encrypted cloud data. Research paper, „Semantic-aware Searching over Encrypted Data for Cloud Computing‟ published by Zhangjie Fu, [1] ,in this to address the problem of semantic retrieval, author propose effective schemes based on concept hierarchy. To improve accuracy, author extends the concept hierarchy to expand the search conditions. Paper, "Towards Efficient Content-aware Search over Encrypted Outsourced Data in Cloud‟ published by Zhangjie Fu, [2] in this paper, author uses an new semantic search scheme based on the concept hierarchy and the semantic relationship in concepts in the encrypted datasets. More specifically, our scheme first indexes the documents and builds trapdoor basedontheconcepthierarchy.Paper,„Dual-ServerPublic-KeyEncryption with Keyword Search for Secure Cloud Storage ‟published by R. Chen, [3] the searchable encryption which allows the user to retrieve the encrypted documents that contain the user-specified keywords, where given the keyword trapdoor, the server can find the data required by the user without decryption. Author investigates the security of a well-known cryptographic primitive, namely Public Key Encryption with Keyword Search (PEKS) which is very useful in many applications of cloud storage. Paper, "Identity- based Encryption with Outsourced Revocation in Cloud Computing‟ published by Jin Li, [4] in this, Identity-Based Encryption (IBE) which simplifies the public key and certificate management at Public Key Infrastructure (PKI) is an another relevant to public key encryption. Paper, "Privacy-Preserving Smart Semantic Search Based on Conceptual Graphs Over Encrypted Outsourced Data‟ published by Zhangjie Fu [5], in this, Considering various semantic representation tools, author select Conceptual Graphs as our semantic bearer because of its great ability of expression and extension. To improve the efficiency of retrieval, author uses Tregex simplify the key sentence and make it more generalizable. Here transfer of CG into its linear form with some alteration which makes quantitative calculation on C Gandfuzzy retrieval in semantic level possible. Paper, "Secure kNN Computation on Encrypted Databases‟ published by, W. K. Wong [6] in this, author discuss the general problem of secure computation on an encrypted database and propose a SCONEDB (Secure Computation ON an Encrypted Database) model, which captures the execution and security requirements. Author focus on the problem of k-nearest neighbor (kNN) on encrypted datasets. Paper, "Building and Applying a Concept Hierarchy Representation of a User Profile‟ published by, Nikolaos Nanas
[7] In this, author creates method for the construction of a concept hierarchy that takes three basic dimensions of term dependence. Paper,„WordNet: A Lexical Database for English‟ published by, George A. Miller[8] in this, Word Net provides a more effective

combination of traditional lexicographic information and modern computing. Word Net is an online lexical database design to use under program control. Paper, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing" published by, JinLi [9] in this, author exploit edit distance to quantify keywords similarity and develop leading technique while constructing fuzzy keyword sets, which reduces the storage. Paper, "Privacy Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data" published by, Ning Cao [10] in this, author define and solve the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing(MRSE)

III. EXISTING

Searchable encryption schemes usually generate a searchable index based on the keyword dictionary, which is extracted from the outsourced dataset, and upload the encrypted index together with encrypted dataset to the cloud server. With the trapdoor generated in the search stage, the server can search the searchable index and return related documents. Traditional searchable encryption schemes only supportsing lekeywordsearchandtake inverted index as its index structure. In order to improve the functionality and usability of the search system, some works are focused on fuzzy keyword search, similarity search and ranked search. Scheme in searchuseseditdistancetoextendkeyworddictionarytop rovidefuzzykeywordsearch. By utilizing keyword weight and order preserving encryption technique, schemes can rank search results and return most relevant documents. Semantic search becomes increasingly important and more and more researchersengagedinthefield,astraditionalkeywordbas edsearchscheme cannot exploit the hidden meanings of terms and the semantic similarity between terms. The concept hierarchy, a semantic tool used for organizing concepts, is mainly constructed to indicate the relationships between concepts. The most important usage of concept hierarchy is to distinguish meanings for classification. The basic idea to define the semantic distance betweentwoconceptsisbasedonthenumberofarcsinthes hortestpathsof two concepts in the concept hierarchy. The schemes are less efficient than the scheme in this paper, because the construction of concept graph is more complex. Compared with these schemes, our proposed methods keep the balance between the efficiency and thesemantics.

DISADVANTAGES

- Efficiency islow
- accuracy islow

IV. PROPOSED WORK

In this paper, we propose an efficient searchable encrypted scheme based on concept hierarchy supporting semantic search with two cloud servers. A concept hierarchy tree is constructed based on domain concepts related knowledge of the outsourced dataset. We extend concept hierarchy to include more semantic relations between concepts. With the help of extended concept hierarchy, document features are extracted moreprecisely and search terms are well extended based on the semantic relationsbetweenconcepts. For each document, two index vectors are generated, one is used tomatchconceptsinthesearchrequestandanotheroneisu sedtodetermine whether the value for an attribute is satisfied with the search request. Correspondingly, the search trapdoor for a search request also includes two vectors.Thereasonwhywechoosetwocloudserversistha ttwoserverscan savemuchtimeinsearch.Oneisusedtocomputethesimila ritybetweenthe documents vector and the trapdoors vector. Another one is used to rank results and return them to users.

ADVANTAGES

- More Accuracy andEfficiency
- Moresecure

Related work:

Early searchable encryption (SE) schemes provide the solution mainly for secure exact keyword search [[3]-[8]]. In the symmetric key setting, Song et al. proposed the first SE scheme, where each word in the file should be encrypted with a two-layered encryption construction independently [[3]]. To improve search efficiency, some researchers turn to index technique. Goh et al. and Chang et al. both proposed similar secure per-file index, where an index including trapdoors of all unique words is constructed for each file [[4],[6]]. Curmola et al. presented a per-keyword index construction, where each entry of the whole hash table index contains the trapdoor for a keyword and an encrypted set of file identifiers [[7]]. To further enhance system usability, some other researchers propose ranked search. Wang et al. proposed a solution for ranked single-keyword search regarding to certain relevance score [[9],[10]]. Cao et al. and Yang et al. proposed the scheme for multi-keyword ranked search, where "Inner product similarity" is used for result ranking [[11],[12]]. Emil et al proposed a hierarchical index structure to achieve more secure and effective dynamic updating [[13]]. As a complementary approach, Boneh et al. proposed the first public key based searchable encryption scheme in the public-key setting [[5]].

However, all the above schemes support only exact keyword search. Namely, users' searching input should exactly match the keywords contained in the files. As an attempt to enhance search flexibility, fuzzy keyword search over encrypted cloud data has been proposed [[14]-[16],[19]]. Li et al. and Wang et al. both exploited edit distance as the similarity metric of keywords to construct the fuzzy keywords set as indexes. Besides, the wildcard-based technique is used for storage-efficiency of fuzzy keywords set [[15],[14]]. Liu proposed "dictionary-based fuzzy set construction" to further reduce the size of fuzzy keywords set [[17]]. Relying on an asymmetric security model, Bringer et al. proposed a fuzzy search scheme based on the embedding of edit distance into Hamming distance [[19]]. This scheme does not need priori define of fuzzy keywords set. Chuah proposed a fuzzy multi-keyword search scheme, where edit distance is also used to evaluate the similarity between terms [[16]]. Besides, an index BedTree is constructed to improve search efficiency with n-gram technique. Without the construction of fuzzy keywords set, Jin introduced new measures, e.g. n-gram bloom-filter and frequency vector, to approximately measure the similarity over encrypted string [[18]]. Note that, the above fuzzy search systems consider the similarity metric mainly from the structure of keywords, not from the semantic relationship. Thus, practically usable semantic search remains to be addressed in the context of encrypted data search.

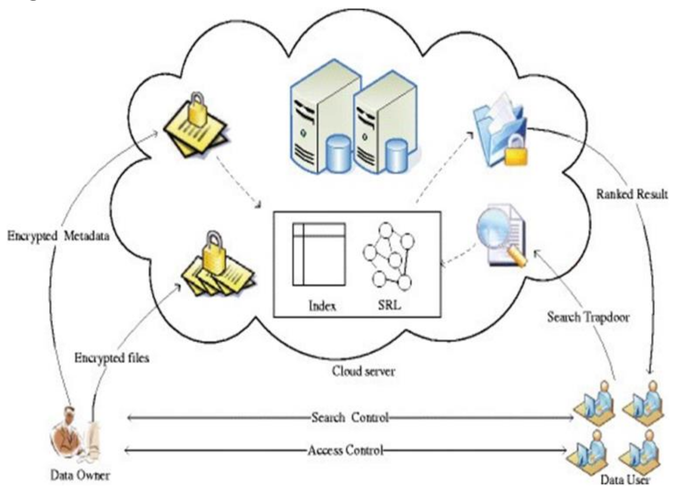
In this paper, we propose a ranked semantic expansion based similar search scheme in the symmetric key setting, which take both the semantic search and result ranking into consideration.

Problem formulation

System model

We consider the system model involving three different entities: data owner, data user and cloud server, as illustrated in Figure 1.

Figure 1



Framework of the semantic expansion based similar search over encrypted cloud data.

Data owner uploads a collection of n text files $F = \{F1, F2, F3, \cdots, F_n\}$ in encrypted form C, together with the encrypted metadata set, to the cloud server. Note that, a corresponding file metadata is constructed for each file. Each file in the collection is encrypted with common symmetric encryption algorithm, e.g. AES.

Data user provides a search trapdoor T_w for keyword w to the cloud server. In our paper, we assume the authorization between the data owner and users is appropriately done.

Cloud server first constructs the index and SRL using the metadata set provided by data owner, thus reduce the computing burden on owner, e.g. index creating. Upon receiving the request T_w, the cloud server automatically expands the query keyword based on SRL. Then the server searches the index, and returns the matching files to the user in order. Finally, the access control mechanism, which is out of the scope of this paper, is employed to manage the capability of the user to decrypt the received files.

Threat model

In this paper, we use the same threat model described in previous searchable symmetric encryption (SSE) scheme [[6],[7],[9],[11],[15],[16]]. We consider an "honest-but-curious" server in our model. Specifically, the cloud server honestly follows the designated protocol specification, but is "curious" to infer and analyze all data information available on the server so as to learn additional information. In other words, the cloud server has no intention to actively modify the stored data or disrupt any other kind of service. Thus we consider the threat models with attack capabilities as follows.

Known background Model: In this model, except for the encrypted dataset and metadata set the owner upload, the server is assumed to have additional knowledge on the dataset, e.g. the subject and its related statistical information. For instance, the server can utilize the keyword frequency statistics to infer keywords.

Design goals

To enable effective and secure ranked semantic expansion search over outsourced cloud data under the aforementioned model, our mechanism should achieve the following design goals.

1. Ranked semantic expansion search: To design a similar search scheme that supports semantic search over encrypted cloud data by expanding the query keyword upon semantic relationship of terms, which finally returns the retrieved files in order.
2. Security guarantee: To prevent cloud server from learning the plaintext of the data files and keywords. Compared to the existing SSE schemes, the scheme should achieve the as-strong-as possible security strength.
3. Efficiency: To achieve the above goals with minimum communication and computation overhead.

Notation

F – the plaintext file collection, denoted as a set of n data files $F = \{F1, F2, \cdots, F_n\}$.

C – the encrypted file collection, stored in the cloud server, denoted as $C = \{c1, c2, \cdots, c_n\}$.

$id(F_i)$ – the identifier of file F_i that can help uniquely locate the actual file.

W – the dictionary, i.e., the keywords set extracted from F, denoted as a set of m keywords $W = \{w1, w2, \cdots w_m\}$.

M – the encrypted metadata set, denoted as a set of n file metadata $M = \{M(F_i)\}, i = 1, 2, \cdots n$.

I – the inverted index including a set of m posting lists $I = \{I(w_i)\}, i = 1, 2, \cdots m$.

T_w – the trapdoor generated for a query keyword w by a user.

S_w – the semantically expanded keyword set of w, it is a subset of W, denoted as Sw={w'1,w'2,⋯}Sw=w1',w2',⋯.

Preliminaries

Semantic query expansion

In the domain of plaintext retrieval, automatic query extension has been a technique to improve the recall and precision of retrieval for a long time [[20]]. It uses the semantically related words to expand the particular query, and makes the query request more satisfy the user's intent.

The key step of semantic query expansion is to find out the semantic relationship between the keywords. Some researchers utilized readily available corpus independent knowledge models [[21]], e.g. WordNet, EuroWordNet, and some others dynamically constructed the semantic relationship from the document collection by the technologies such as term clustering [[22],[23]], and mutual information model [[24]-[26]]. Among these technologies, mutual information model is widely used [[24],[26]-[29]].

Refer to the formula used in [[26]], which adopted the mutual information model to implement semantic search in web. The mutual information I(x, y) is defined as

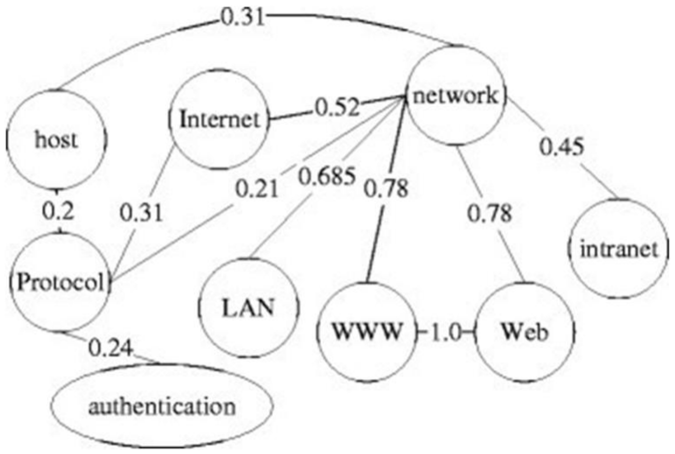
I(x,y)≡log2P(x,y)p(x)p(y)Ix,y≡log2Px,ypxpy

(1)

Here P(x, y) is the probability of observing x and y together. p(x) and p(y) are the probabilities of observing x and y independently in the collection. The higher the semantic relationship between x and y is, the larger the co-occurrence degree is, and consequently the larger the mutual information I(x, y) is.

Then normalize the mutual information into a value of relationship in interval [0, 1]. The semantic relationship library will be constructed as a weighted graph structure showed in Figure 2.

Figure 2



An example of semantic relationship library.

Inverted index

Inverted index is a widely used indexing structure in information retrieval. It is consist of a list of mappings from keywords to the set of files that contain this keyword [[30]]. For the purposes of ranking, the numerical relevance score is computed for each file based on TF × IDF rule introduced later in subsection "Basic definition". An example index structure of keyword w i is shown in Table 1. Here S ij (j = 1, ⋯, n i) denotes the relevance score of file F ij in response to w i , n i is the number of files contain keyword w i .

Table 1 An example of inverted index

Order-preserving Encryption (OPE)

The OPE is a deterministic encryption scheme, whose encryption function preserves the numerical ordering in plaintext-space [[31],[32]]. More specifically, a function $f : D = \{1, \cdots, M\} \rightarrow R = \{1, \cdots N\}$ is order-preserving, if for all $a, b \in D$, $f(a) > f(b)$ if $a > b$. Generally, any order-preserving function can be defined as a combination of M out of N ordered items, which can be calculated by (NM)NM. The adversary has to execute exhaustive enumeration, namely searching over all the possible combination, to break the encryption. So the number of combination, which is maximized when $M = N/2$, should be large enough to ensure the security. If the security level is chosen to be 280, since (N/M)M≤(NM)N/MM≤NM, it is suggested to choose $M = N/2 > 80$. A plaintext m in domain D is always mapped to a random-sized non-overlapping bucket in range R. Then a ciphertext c is chosen within the bucket depend on the value of some random function.

Basic definitions

Ranking function

A ranking function is used to measure relevance scores of matching files to a given query in information retrieval. The most widely used measurement for evaluating relevance score is TF × IDF rule. TF (Term frequency) is used to measure the importance of the term within the particular file, defined as the number of times a given term or keyword appears within a file. IDF is used to measure the overall importance of the term within the whole collection, defined as the total number of documents in the collection divided by the total number of documents including that word. Note that, we focus on single keyword search in our scheme. Thus without loss of generality, the relevance score of single keyword can be computed using equation 2, which is widely used in the literature [[33]]:

Score(w,Fi)=1|Fi|·(1+lnfi,w)·ln(1+nfw)Scorew,Fi=1Fi·1+lnfi,w·ln1+nfw

(2)

Here w denotes the query keyword; fi,w is the TF of term w in file F i ; f w denotes the number of files that contain keyword w. n is the number of files in the collection, while |F i | is the length of file F i , obtained by counting the number of indexed terms in the file.

In our scheme, we first expand the query keyword based on SRL, and then both the keyword and its semantically related words are used to retrieve the files. So F d 's total relevance score will be computed for result ranking with equation 3.

TScore(w,Fd)=Scorew+∑∀wi'∈SwScorew'i×RiTScore w,Fd=Scorew+∑∀wi'∈SwScorewi'×Ri

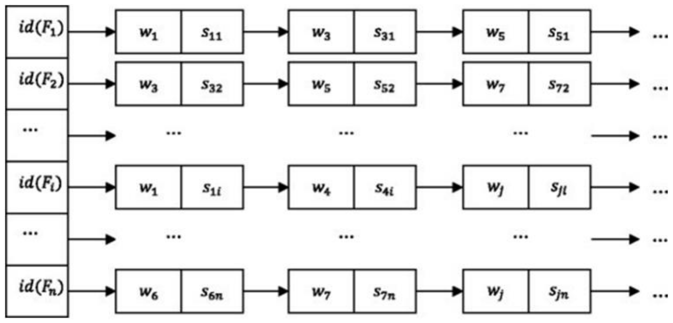
(3)

Here Score w represents the relevance score of the input keyword; Scorew'iScorewi' represents the relevance score of expanded keyword w'iwi', while R i is the value of semantic relatedness.

File metadata

A piece of file-metadata is constructed for each file. The file-metadata consists of the file ID, keywords, and the relevance scores (refer to equation 2) of keywords in response to the file. If file F i contains keyword w j , a tuple w j , s ji is insert into metadata M(F i), where s ji represents the relevance score of keyword w j response to file F i . All of the file metadata constitute metadata set, which is shown in Figure 3.

Figure 3


$id(F_1)$
w_1 s_{11} w_3 s_{31} w_5 s_{51} …
$id(F_2)$
w_3 s_{32} w_5 s_{52} w_7 s_{72} …
…
$id(F_i)$
w_1 s_{1i} w_4 s_{4i} w_j s_{ji} …
…
$id(F_n)$
w_6 s_{6n} w_7 s_{7n} w_j s_{jn} …

An example of metadata set.

Secure Semantic Expansion based Similar Search Scheme

The scheme consists of six algorithms (KeyGen, BuildMD, BuildIndex, BuildSRL, TrapdoorGen, and SearchIndex), which can be constructed in two phases—Setup and Retrieval.

The setup phase

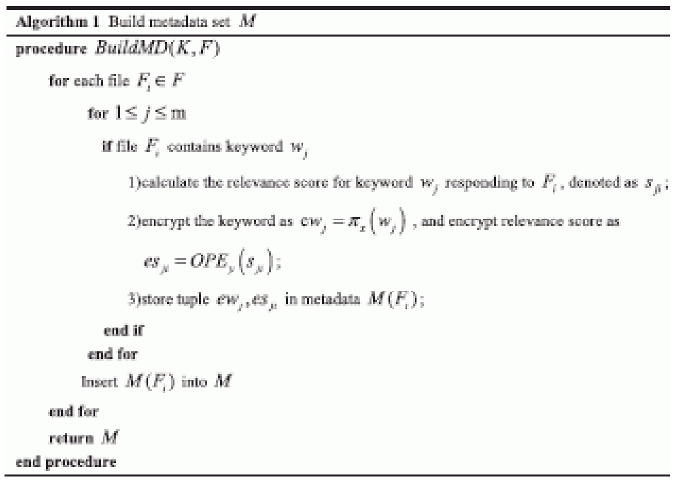
In this phase, data owner initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the file collection F using BuildMD to generate the encrypted metadata for each file. Finally the owner uploads both the encrypted file collection C and metadata set M to the cloud server. With M received from data owner, the server constructs the index using BuildIndex and semantic relationship library using BuildSRL. In addition, the necessary secret parameters, e.g. the trapdoor generation key, should be distributed to a group of authorized users by employing off-the-shelf public key cryptography or broadcast encryption. Details are as follows:

1. 1)

The data owner initiates the scheme by calling KeyGen(1k, 1l, 1P). It takes the security parameters k, l, p as inputs and generates random keys x←R{0,1}kx←R0,1k, y←R{0,1}ly←R0,1l. Finally it outputs secret keys set K = {x, y, 1l, 1P } used for later encryption, such as trapdoor generation and relevance score encryption.

2. 2)

Then the data owner builds the secure metadata for each file in file collection F by calling BuildMD(K, F), It takes the secret K and dataset F as inputs and outputs the encrypted metadata set M. The function extracts the keywords in each file and computes the corresponding relevance score. The keyword in the metadata is encrypted with collision resistant hash function π: {0, 1}k × {0, 1}* → {0, 1}p (p > log m), where m denotes the size of keywords set. The relevance score is encrypted with order-preserving encryption algorithm OPE: {0, 1}l × {0, 1}d → {0, 1}r, where d and r respectively represent the bit length used to denote all the values in domain D and range R. The detail is shown in Algorithm 1.

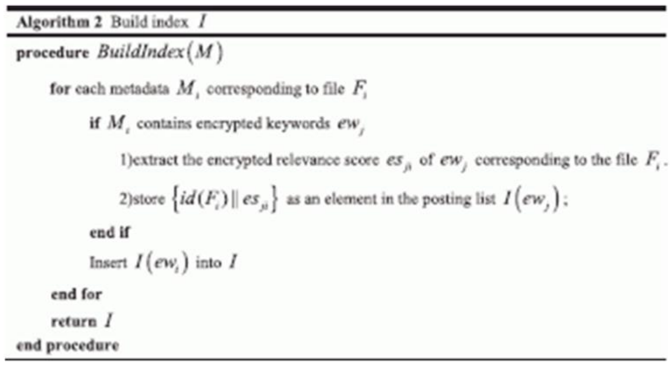


```
Algorithm 1  Build metadata set $M$
procedure $BuildMD(K, F)$
    for each file $F_i \in F$
        for $1 \le j \le m$
            if file $F_i$ contains keyword $w_j$
                1)calculate the relevance score for keyword $w_j$ responding to $F_i$, denoted as $s_{ji}$;
                2)encrypt the keyword as $ew_j = \pi_x(w_j)$, and encrypt relevance score as
                  $es_{ji} = OPE_y(s_{ji})$;
                3)store tuple $ew_j, es_{ji}$ in metadata $M(F_i)$;
            end if
        end for
        Insert $M(F_i)$ into $M$
    end for
    return $M$
end procedure
```

Figure 4 is an example of the encrypted metadata set.

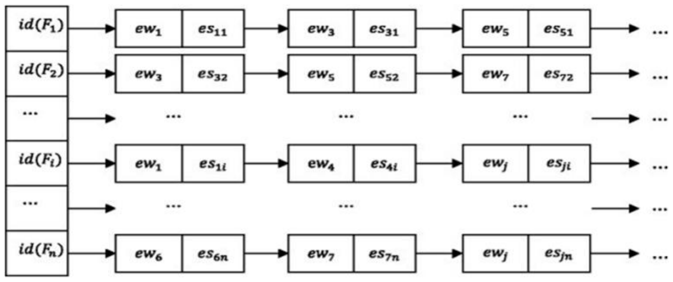
3. 3)

When receiving the secure metadata, the server builds the inverted index by calling BuildIndex(M). The function extracts the encrypted keywords and constructs a posting list for each keyword. If keyword ew j included in file metadata M(F i), the element {id(F i)||es ji } is inserted into posting list of keyword ew j . The details are given in Algorithm 2. The SRL is also built upon the metadata set and uses common association rules algorithm to mining the co-occurrence relationship of keywords.



```
Algorithm 2  Build index $I$
procedure $BuildIndex(M)$
    for each metadata $M_i$ corresponding to file $F_i$
        if $M_i$ contains encrypted keywords $ew_j$
            1)extract the encrypted relevance score $es_{ji}$ of $ew_j$ corresponding to the file $F_i$.
            2)store $\{id(F_i) \| es_{ji}\}$ as an element in the posting list $I(ew_j)$;
        end if
        Insert $I(ew_j)$ into $I$
    end for
    return $I$
end procedure
```

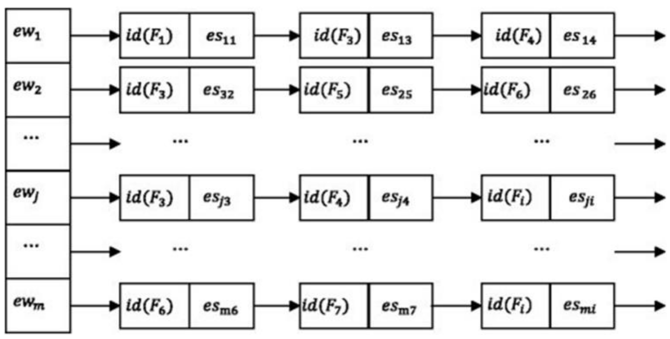
Figure 4


id(F1) ew1 es11 ew3 es31 ew5 es51
id(F2) ew3 es32 ew5 es52 ew7 es72
id(Fi) ew1 es1i ew4 es4i ewj esji
id(Fn) ew6 es6n ew7 es7n ewj esjn

An example of encrypted metadata set.

An example of secure inverted index constructed by cloud server is shown in Figure 5.

Figure 5


ew1 id(F1) es11 id(F3) es13 id(F4) es14
ew2 id(F3) es32 id(F5) es25 id(F6) es26
ewj id(F3) esj3 id(F4) esj4 id(Fi) esji
ewm id(F6) esm6 id(F7) esm7 id(Fi) esmi

An example of secure inverted index.

The retrieval phase

In this phase, the user generates a secure trapdoor of his interested keyword using TrapdoorGen, and submits it to the cloud server. Upon receiving the query trapdoor, the cloud server first automatically expands the query keyword. Then the server searches the index via SearchIndex, and eventually sends back the matched files in a ranked sequence according to the total relevance scores. During the process, beyond the order of the relevance scores, nothing or little information should be leaked. Details are as follows.

1. The user generates a trapdoor $T_w = \pi_x(w)$ for an interested keyword w, by calling TrapdoorGen(w).
2. Upon receiving the trapdoor T_w, the server first expands the query keyword to obtain the extensional query trapdoor $T_w' = \{\pi_x(w), \pi_x(w_i')\}, \forall w_i' \in S_w$. By calling SearchIndex, the server locates the matching entries of the index via $\pi_x(w)$ and $\pi_x(w_i')$, which include the file identifiers and the associated order-preserved encrypted relevance scores.
3. The server then computes the total relevance score of each file to the query according to equation 3. In the end, the server sends back the matched files in a ranked sequence, or sends top-k most relevant files if the user provides the optional value k.

Towards one-to-many order-preserving encryption

To implement efficient result ranking, we use OPE encrypt the relevance score. Thus the server can rank the retrieved files directly according to the encrypted relevance score. However, the original OPE is a deterministic encryption scheme, if not disposed properly, it will leak as much information as any deterministic encryption scheme does [[32]]. In particular, the statistical information of the scores, such as the distribution slope, value range etc., can be used to identify the specific keyword in the query [[9]].

Therefore we need to modify the OPE to suit our requirement. The original OPE first maps the plaintext m in domain D to an interval bucket in range R. Then the ciphertext c is chosen in the bucket using m as the random seed for the random selection function. The modified OPE should map the same plaintext score to different ciphertext, and still globally preserve the order of relevance score. Thus a one-to-many OPE scheme is desired to reduce the amount of information leakage. More specifically, in the final ciphertext selection process, together with the plaintext m, the unique file ID is introduced as an additional random seed. Thus the same plaintext will not be deterministically mapped to the same ciphertext, but a random value within the randomly assigned bucket in range R. Algorithm 3 shows the whole process, where GetCoins(·) is a random coin generator, HYGEINV(·) is the HGD(·) sampling function instance in MATLAB. In the process, a plaintext m in domain $D = \{1, \cdots, M\}$ is mapped into ciphertext c selected in range $R = \{1, \cdots N\}$, id(F) denotes the corresponding file ID. In the paper, the one-to-many OPE is denoted as OM – OPE.

Algorithm 3 one-to-many Order-preserving encryption

```
procedure $OM-OPE_k(D,R,m,id(F))$
    while $|D|!=1$ do
        $\{D,R\} \leftarrow BinarySearch(K,D,R,m)$;
    end while
    $coin \overset{R}{\leftarrow} GetCoins(K,(D,R,1||m,id(F)))$;
    $c \overset{coin}{\leftarrow} R$;
    return $c$
end procedure
procedure $BinarySearch(K,D,R,m)$
    $M \leftarrow |D|; N \leftarrow |R|$;
    $d = min(D)-1; r = min(R)-1$;
    $y \leftarrow r + N/2$;
    $coin \overset{R}{\leftarrow} GetCoins(K,(D,R,0||y))$;
    $x \overset{R}{\leftarrow} d + HYGEINV(coin,M,N,y-r)$;
    if $m \le x$ then
        $D \leftarrow \{d+1,\cdots,x\}; R \leftarrow \{r+1,\cdots,y\}$;
    else
        $D \leftarrow \{x+1,\cdots,d+M\}; R \leftarrow \{y+1,\cdots,r+N\}$;
    end if
    return $\{D,R\}$;
end procedure
```

The mapping scheme should be as random as possible to eliminate the predictability of the keyword specific score distribution. Obviously, the larger the size of range R is, the less specific characteristics will be preserved. However, considering the efficiency of HGD function, the size of range R cannot be unboundedly large. So the range size |R| should be properly tradeoff between randomness and efficiency. To guarantee the security of keywords in the metadata set, the relevance score should be encrypted with OM – OPE y (·) instead of OPE y (·) in Algorithm 1.

Security analysis

We estimate the security of the proposed scheme by proving the security guarantee stated above (refer to Design goals). That is, both the data files and the keywords are not leaked to the server.

Security analysis for the ranked semantic expansion Search

We analyze the solution with respect to the aforementioned search privacy requirement, e.g. keyword privacy and file confidentiality.

- **File confidentiality:** the file confidentiality depends on the inherently security strength of the symmetric encryption scheme, so the file content is obviously protected well.
- **Keyword privacy:**

1. 1.

The query trapdoor is generated using the symmetric encryption scheme, so the privacy of query keyword depends on the inherently security strength of the symmetric encryption scheme.

2. 2.

The proposed scheme introduces some additional information in the index compared to the original SSE, such as the encrypted relevance scores and the values of relationship between terms. Thus the privacy of keyword in the index depends on not only the symmetric encryption scheme. We discuss the security from two aspects.

On one hand, as defined in the thread model, the server may predict the plaintext of keyword depends on the score distribution. Thus the OM – OPE is used to encrypt the score, which could flatten the distribution of relevance score. So the keyword privacy mainly depends on the security of OM – OPE. In the next part, we analyze the security of OM – OPE in detail. As discussed, if the data owner properly enlarges the range R, the relevance score will be randomly mapped to a sequence of order-preserved numeric values with very low duplicates. So OM – OPE makes it difficult for the adversary to predict the plaintext score distribution, let alone predict the keywords.

On the other hand, as shown in Table 2, the semantic relationship values between terms do not have their peculiarities, which cannot be effectively used for statistical analysis. Note that, in the previous literature with inverted index [[9]], the server can also get the co-occurrence degree of terms by recording and analyzing the search result. Thus the leaking of relationship information shouldn't be a main secure problem we have to solve in current work.

Table 2 An example of semantic relationship between terms

Security analysis for one-to-many OPE

The one-to-many OPE scheme introduces the file ID as the additional seed in the ciphertext chosen process. So the same plaintext will not be deterministically mapped to the same ciphertext, but a random value in the assigned bucket in range R. This helps flatten the score distribution of keyword, and protect the keyword privacy from statistical attack.

However, if there are many duplicates of plaintext m, the ciphertext distribution may not be flattened effectively for the small size of assigned bucket in range R. So we should expand the range R properly to ensure the low duplicates on the ciphertext range, it will be difficult for the adversary to analyze which points in R belong to the same plaintext score.

In this paper, we use the min-entropy to choose the size of R. It is defined as: H(σ)=–log(maxαPr[σ=α])Hσ=–logmaxαPrσ=α, where σ is a discrete random variable, α denotes a state of σ with the max probability. In general, the higher H(σ) is, the more difficult the σ can be predicted. If H(σ) ∈ w(log k), the min-entropy of variable σ will be high, where k is the bit length needed to denote all the states of σ [[8]]. We could choose H(σ) as (log k)c where c > 1 [[9]]. Then the least size |R| should satisfy the equation 4:

(log(log|R|))C≤–log$\Bigg(\Bigg|$ max/(|R|·125logM+12)δ $\Bigg)\Bigg)$ |l
oglogRC≤–logmax/R·125logM+12δ
(4)

Here max denotes the maximum number of score duplicates within the metadata set. δ denotes the totoal number of scores to be mapped within metadata set. Wit D = {1, ⋯, M}, M = |D|, the total recursive calls of BinarySearch(·) function (line 9 in Algorithm 3) is at most 5 log M + 12 on average. If the range size |R| is denoted in bits, namely k = log |R|, we will get equation 5. With the established file metadata set, it is easy to determine the proper rage size |R|.

max·25logM+122k·δ=max·M52k–12·δ≤2–(logk)cmax·
25logM+122k·δ=max·M52k–12·δ≤2–logkc
(5)

As discussed above, if we properly choose the range R, the randomness in the ciphertext selection process will effectively mitigate the useful information revealed to the cloud server.

Performance analysis

To evaluate the performance of our proposed scheme, we implemented the secure search system using C++ on a windows machine with Intel Core 2 Duo CPU Processor running at 2.93GHZ, 2.94GHZ. The experimental evaluation was conducted on a real data set: Request for comments database(RFC) [[34]], this file set contains a large number of technical keywords. The overall performance evaluation of our scheme includes the cost of metadata construction, the time necessary for index and SRL construction as well as the efficiency of search.

Metadata construction

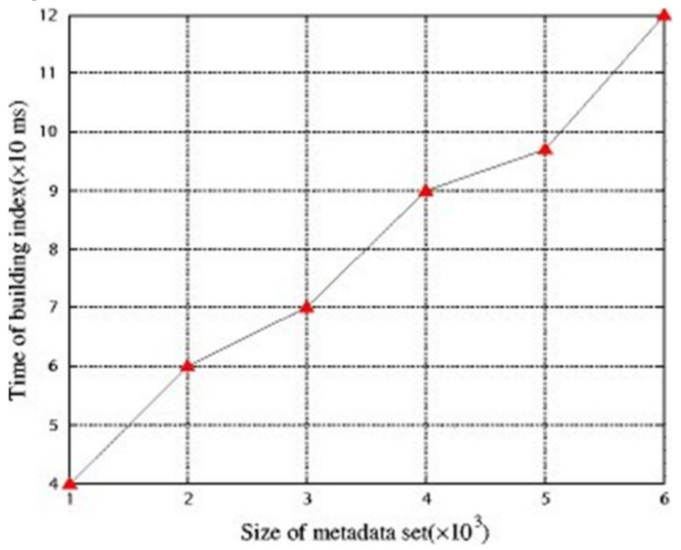
The main overheads for data owner are time cost and storage cost of metadata construction. To build a metadata for each document F i in the dataset F, we should extract the keywords and compute the associated relevance score, then encrypt the keywords and scores. The time cost of each entry directly depends on the number of keywords in the file, while the overall efficiency is also related to the number of the files in the collection. So Table 3 lists the metadata construction performance for a dataset of RFC files. Both the metadata size and construction time listed are the average value, for the reason that it eliminates the difference of various file set construction choices.

Table 3 File metadata construction overhead

Index and SRL construction

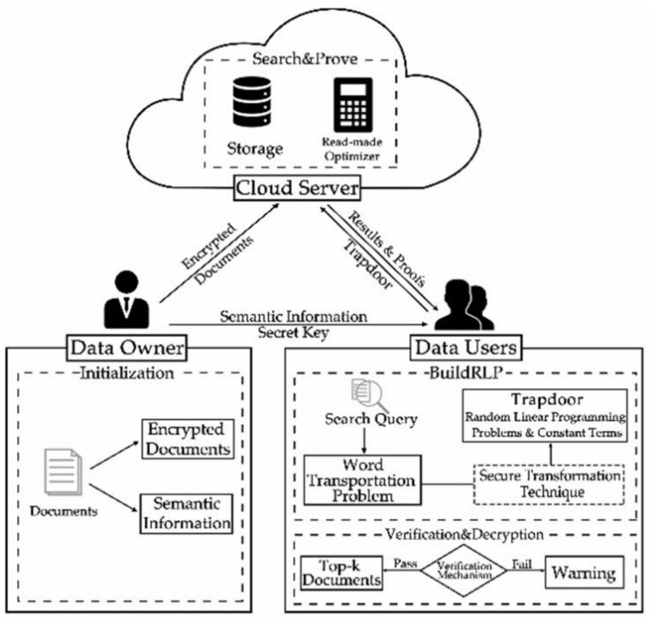
In our construction we should scan the whole metadata set to extract the keywords and build the inverted index with corresponding scores. Figure 6 shows that the whole index is nearly linear with the size of M, namely the number of documents in the collection. The SRL is also built by scanning the metadata set, with the certain support threshold, the number of entries is the main factor to the efficiency. Figure 7 shows the time cost of building SRL against the increasing size of M or dataset. In addition, taking into account the abundant computing resources on server, the performance of building index and SRL is practically efficient.

Figure 6



The time cost for building index.

Figure 7

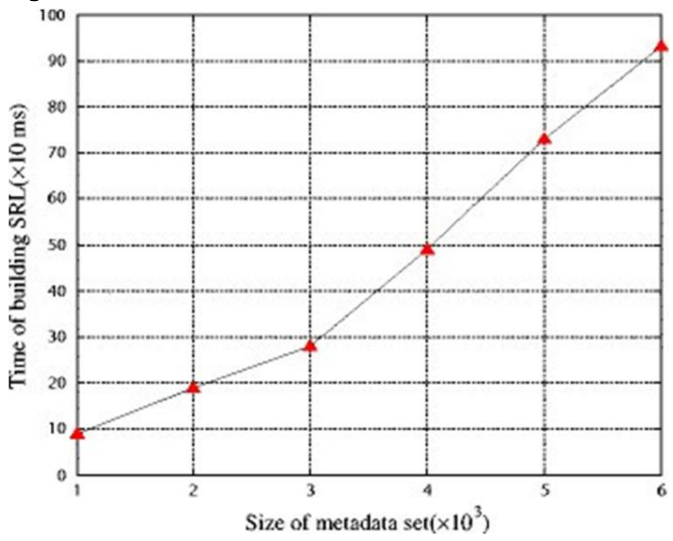


The time cost for building SRL.

search efficiency

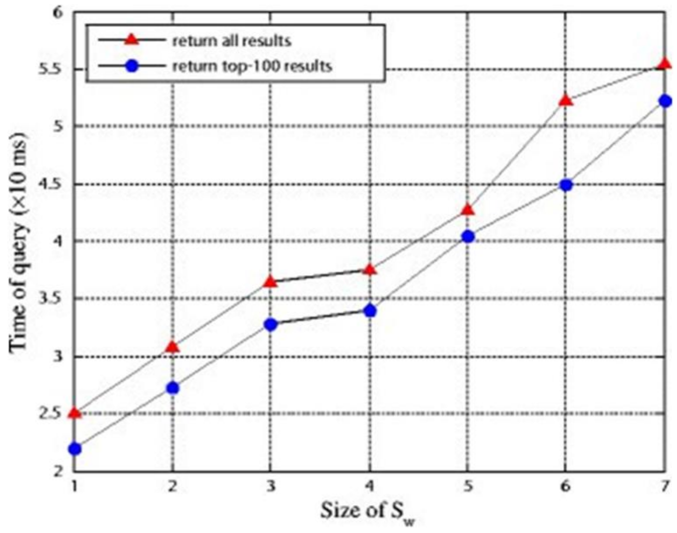
The search process includes query extension, fetching the posting list in the index, calculating the total relevance score and ranking the result in descending order. Compared to the original ranked search, our approach introduces the keyword extension cost, and the calculation cost of final relevance score. So the size of semantically expanded keywords set is a factor to the query efficiency. Figure 8 shows the average time cost of query against the size of S w . With result ranking, top-k search could return the most satisfied files more efficiently. In addition, as the evaluation of overall search performance, Figure 9 shows the average time cost of query against the number of files. Besides, the index and SRL could be stored with a tree based data structure, so that the server does not need to traverse all the keywords entries.

Figure 8

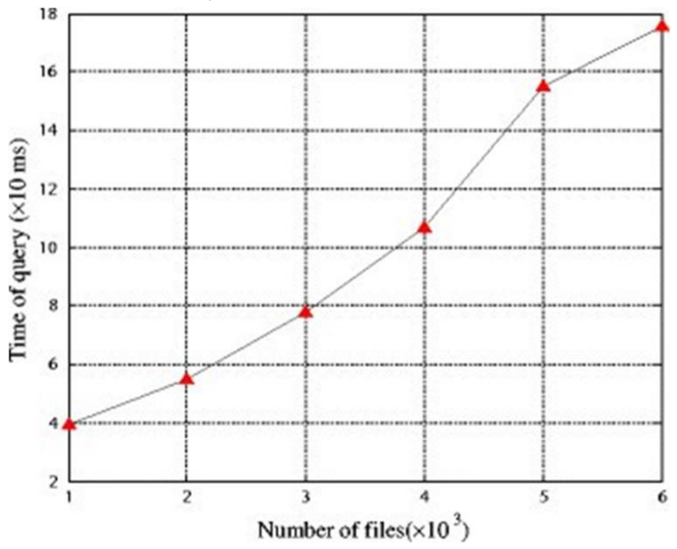


Time cost of query. For query keyword with different size of semantically expanded keywords set, n=1000.

Figure 9



The overall query performance.



Recall factor of the search

By analyzing the search result, the overall recall rate is improved, and the query results are more in line with the user's actual intentions. E.g. a user inputs a keyword 'protocol', the files which contain related words like 'internet', 'network', 'authentication' will also be returned, in addition, the files which include most of the words will also be ranked forward.

V. STRUCTURE DIAGRAM

Figure 1. The system architecture of our secure verifiable semantic searching scheme.

As illustrated in Fig. 1, there are three entities involved in our system: the data owner, data users, and the cloud server.

The data owner has a lot of useful documents, but only haslimited resources on the local machines. Therefore, the owner

highly motivated to perform Initialize () for initializing the

scheme. The owner encrypts documents Fto get documents Cwith secret key K, then outsources Cto the cloud server. The data owner builds forward indexes I, then sends indexes IandKto data users.

Data users are the searching that send the trap-door of a query to the cloud server for acquiring top-krelated documents. Specifically, users input arbitrary query words q,

perform BuildRLP () to generate word transportation problems Ψ, after transform Ψ to random linear programming problems Ω and the corresponding constant terms Δ as a trap- proofs Λ returned from the cloud. Users perform VerDec () to documents when Λ passes our verification mechanism. . Afterward, users receive top-kencrypted documents and The cloud server is an intermediate service provider that

the encrypted document Cand performs the retrieval process. Once receiving the trapdoor, the cloud server

SeaPro () for leveraging any ready-made optimizer to solve the Ω, then obtains the encrypted minimum wordtransportation cost values with Δ. The cloud ranks the valuesin ascending order and returns the top-kencrypted documents
users. In the process, the cloud server also provides proofs
Λ for proving the correctness of the search results.

VI. CONCLUSION

In this paper, as an initial attempt, we propose a secure semantic expansion based similar search scheme over encrypted cloud data. The proposed scheme could return not only the exactly matched files, but also the files including the terms semantically related to the query keyword. The encrypted files and metadata set are outsourced to the server by the owner. With the file metadata, the cloud builds the inverted index and constructs semantic relationship library (SRL) for the keywords. The co-occurrence of terms is used to capture the semantic relationship of keywords in the dictionary, which offers appropriate semantic distance between terms to accomplish the query keyword extension. Then we derive a one-to-many OPE scheme to protect the term frequency, while ensure the computing of total relevance score. Experimental evaluation demonstrates the efficiency and effectives of the scheme.

As our future work, the most practical one is to further improve the security of our solution. Thus new crypto techniques still need to be designed to protect the semantic information while keep the ability to calculate the relevance score. In addition, we intend to research on multi-keyword semantic search scheme which further introduces the semantic relationship between terms, e.g. the position of terms.

ACKNOWLEDGMENT:'

Thispaper proposed an efficient semantic searching method over encrypted cloud data.We thank Shaocong Wu for valuable discussions and his help in experiments. This work was supported in part by the Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing), in part by Shenzhen International co- h projects GJHZ20170313150021171, and in part by ointed Founding (U1613215).

VII. REFERENCES

[1]. Zhangjie Fu, Lili Xia, Xingming Sun, Alex X. Liu, GuowuXie, "Semantic-awareSearchingoverEncryptedDataforCloudComputing",IEEE Transactions on Information Forensics and Security

[2]. Z.Fu,X.Sun,S.Ji,andG.Xie,"Towardsefficientcontent-aware search over encrypted outsourced data in cloud," Proc. of IEEE INFOCOM 2016,pp.1-9,2016.

[3]. R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang. "Dual-server public-key encryption with keyword search for secure cloud storage," IEEE Transactions on Information Forensics and Security, vol.11,no.4, pp.789- 798,2017.

[4]. J. Li, J. Li, and X. Chen, "Identity-based encryption with outsourced revocation in cloud computing," Computers, IEEE Transactions on, vol.64,no.2,pp.425-437,2015.

[5]. Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang. "Privacy preservingsmartsemanticsearchbasedonconceptualgraphsoverencrypted outsourceddata,"IEEETransactionsonInformationForensicsandSecurity, vol.12,no.8,pp.1874-1884,2017.

[6]. W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on

encrypted databases," in Proc. of SIGMOD, 2009, pp. 139–152

[7]. N. Nanas, V. Uren, and A. D. Roeck, "Building and applying a concept hierarchy representation of a user profile," in Proc. Of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval,2003.

[8]. G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol.38, issue 11, pp. 39–41,1995.

[9]. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. of IEEE INFOCOM"10MiniConference,SanDiego,CA,USA,March2010,pp.1–5.

[10]. N. Cao, C. Wang, and M. Li, "Privacy-preserving multi- keyword ranked search over encrypted cloud data," Parallel and Distributed Systems,IEEE Transactions on, vol.25,no.1, pp.222-233,2014.