Hindawi
Security and Communication Networks
Volume 2018, Article ID 1420930, 10 pages
https://doi.org/10.1155/2018/1420930

Research Article

Semantic Contextual Search Based on Conceptual Graphs over Encrypted Cloud

Zhenghong Wang[1], **Zhangjie Fu**[1,2] **and Xingming Sun**[1]

[1] *School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China*
[2] *College of Information Science and Technology, Jinan University, Guangzhou 510632, China*

Correspondence should be addressed to Zhangjie Fu; wwwfzj@126.com

Received 15 July 2018; Accepted 8 November 2018; Published 2 December 2018

Guest Editor: Xuyun Zhang

Currently, searchable encryption becomes the focus topic with the emerging cloud computing paradigm. The existing research schemes are mainly semantic extensions of multiple keywords. However, the semantic information carried by the keywords is limited and does not respond well to the content of the document. And when the original scheme constructs the conceptual graph, it ignores the context information of the topic sentence, which leads to errors in the semantic extension. In this paper, we define and construct semantic search encryption scheme for context-based conceptual graph (ESSEC). We make contextual contact with the central key attributes in the topic sentence and extend its semantic information, so as to improve the accuracy of the retrieval and semantic relevance. Finally, experiments based on real data show that the scheme is effective and feasible.

1. Introduction

1.1. Background. In 2000, digital storage accounted for only 1/4 of the world's data, and another 3/4 of the information was stored in newspapers, books, and other mediums. But by 2020, digital information will account for 4/5 of global data and will reach 40ZB, which is equivalent to 5200GB of data generated by each person. The consumption of the local storage of the users is too expensive. So in order to save storage costs of data, users usually choose to upload data to the cloud. However, public clouds are not always trusted, so the data always is encrypted before uploading to the cloud servers, which also makes the traditional plaintext search scheme invalidated. Thus, how to better protect and utilize user privacy in cloud computing has become a major research issue in mobile cloud computing.

Searchable encryption of the cloud server has become an important field of investigation in recent years. One of the most popular methods of traditional schemes is keywords-based search. The data owner first extracts the corresponding keywords for the data documents and builds the corresponding index and then outsources the encrypted documents and index to the cloud server. When searching for the encrypted data, the cloud server can match the trapdoor with the encrypted index; then the corresponding data documents are returned to the data user. But, as we know, there are some deficiencies with the above keywords-based schemes, which cannot reflect the user's search intention and the semantic information of the document.

In the keyword-based encryption search schemes, the data owner summarizes a document's content into some keywords, which can make search matching efficient and simple. However, the keyword cannot represent the contents of the data document well; it ignores the semantic information of the document. Thus, the returned search results from cloud server are not always matching with the requirement of the user's query. Although the keywords-based schemes [1, 2] have a semantic extension of the keywords, they still cannot overcome the limitations of the keywords. Thus, we research content-based searchable encryption scheme. The scheme [3] takes into account the central content of the text, which expresses the document content with a topic sentence, then establishes the conceptual graph for the topic sentence, and builds the corresponding encrypted index structure. Unfortunately, the scheme does not consider context-sensitive semantics. Thus, the scheme still has a lot of defects.

Therefore, under protecting the security of user privacy in the cloud environment, in order to improve the relevance of documents information obtained by encrypted search, we proposed a searchable encryption scheme which combined the local features with the context similarity.

1.2. Main Contribution. In the paper, we propose a semantic search encrypted scheme based on conceptual graphs of context (ESSEC). We still extract the central content of the whole document as the index rather than keywords and then construct the corresponding weighted conceptual graph [4] for the topic sentence:

(i) We extend the context-based semantics of the center concept attribute, so that the generated conceptual graph can contain the content information of the document and constructs the semantic network of the conceptual graph, which helps to make search results satisfy the needs of users' retrieval as much as possible.

(ii) The experiments based on real datasets have been implemented, and the experimental contrast diagrams make clear that the two schemes put forward in this paper are effective and feasible.

2. Related Work

Searchable encryption [2, 3, 5] is cryptographic primitives developed for data's encrypted search. The symmetric key searchable encryption scheme was first proposed by Song et al. [6]. Subsequently, the early researchers Golle and Ballard et al. [7–9] have proposed the schemes to support multikeyword search in different application scenarios. Returns related documents based on whether the keywords are contained in the document. However, the earlier proposed schemes are only applicable to small-scale specific types of applications and ignore the semantic information of documents.

Cao et al. [10] first defined and solved the problem of multikeyword classification retrieval on encrypted cloud data (MRSE). In the scheme, Cao creatively uses intrinsic product similarity and coordinate matching to compute the correlation between keywords and files and put forward the two different threat models. The first model is a known ciphertext model and other is the known background model. Then, [11–13] have the further study on the basis of MRSE.

Then, the scholars have put forward many excellent schemes based on semantic searchable encryption [14–18]. Li et al [14] first use the wildcard technology and editing distance to construct a fuzzy semantic keyword set. Fu et al. [15] construct wordnet tree to expand its semantics for keywords. Then, [16] was based on the NLP analysis of the input multiple keywords to obtain the weight of each keyword to represent the interest of the user and expand the semantics by extending the central keyword to improve the efficiency and accuracy. However, taking into account the deficiencies of the keywords, literature [19] constructs a content-based semantic searchable encryption scheme, which uses the semantic representation tool of the conceptual graph to store the content information of the document, thereby implementing semantic retrieval. [18] proposes a verifiable diversity ranking search scheme over encrypted outsourced data. In our scheme, we still use the conceptual graph as our semantic expression tool, but we take into account the contextual semantic content when constructing conceptual graph, in order to construct a semantic network, increasing the retrieval accuracy.

3. Problem Formulation

3.1. System Model and Threat Models. The system model considered in this paper is shown in Figure 1: the data owner, data user, and the cloud server are 3 entities of the system. To keep the data private, before uploading the documents $F = \{F_1, F_2, ..., F_n\}$ to the cloud server, the data owner would encrypt the data $C = \{C_1, C_2, ..., C_n\}$. Meanwhile, to retrieve encrypted data, the data owner needs to generate a searchable encryption index I. In our scheme, we generate a conceptual graph index for encrypted documents. Finally, both index and documents will be encrypted and upload to the cloud.

The data users need to obtain the authorization from the data owner. Then they need to generate request trapdoor (conceptual graph) η for query sentence, which will be encrypted upload to the cloud server. And the cloud server matches the encrypted index I with the encrypted trapdoor η. Finally, the cloud will return the related encrypted documents to the data user. The data user would decrypt the encrypted documents.

In our scheme, we think the cloud server is "honest but curious." In other words, the cloud server can comply with the protocols, but it still hopes to obtain more sensitive information through learning and guessing. In this paper, we only focus on how the cloud can deal with the similarity search over the encrypted data, which is the same as the model adopted by previous literature [10].

3.2. Notations and Preliminaries

3.2.1. Notations

(i) F: the plaintext document dataset, $F = \{F_1, F_2, ..., F_n\}$, and each F_i can be summarized as a CG.

(ii) C: the ciphertext document dataset, $C = \{C_1, C_2, ..., C_n\}$.

(iii) CG: conceptual graph

(iv) Q: the query represented by two vectors and a hash table, defined as a collection $Q = \{Q_1, Q_2, QM_3\}$.

(v) QM: the hash structure in query.

(vi) $F(Q_1)$: the encrypted set of documents in F whose D_1 is similar with Q_1.

(vii) D_{ij}: the index composed of vectors and hash table, which is defined as $D_i = \{D_{i1}, D_{i2}, D_{i3}, M_4\}$.

3.2.2. Preliminaries. **Conceptual Graph:** Sowa first proposed the conceptual graph scheme [20], which is the model of

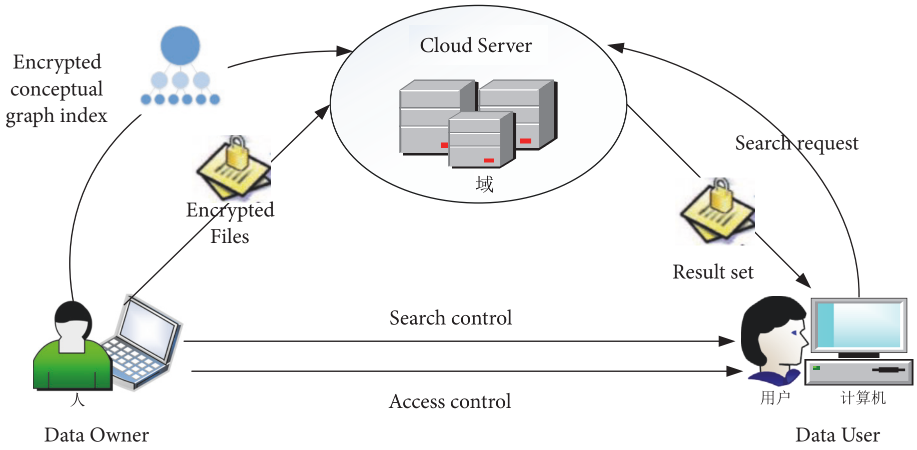


FIGURE 1: System model.

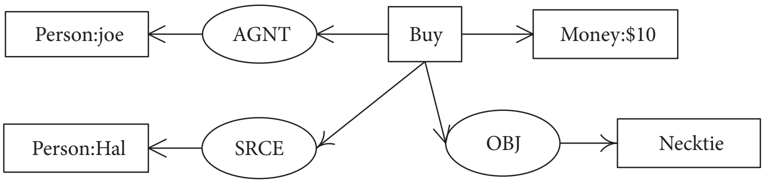


FIGURE 2: Conceptual graph: joe buys a necktie from Hal for $10.

semantic knowledge representation, similar to a knowledge graph. It is the structure of knowledge representation based on first-order logic [21]. As a logical model, conceptual graph can be used to describe any content that can be implemented on the digital computer. It usually has two types of nodes: concepts (rectangles) and conceptual relationships (also known as semantic roles) (Ellipse) (Figure 2). At the same time, for each concept, there are two parts: the left is a type label, which represents the type of entity, and on the right is a concept attribute value, but its concept type does not necessarily exist. Each concept is associated with other concepts. And there are about 30 relationships and 6 tenses.

TF-IDF(term frequency–inverse document frequency): is a statistical method used to reflect how important a word is to a document or corpus [1]. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus. Term frequency (TF) refers to the frequency of a given word in the file. Inverse document frequency (IDF) is a measure of the universal importance of a word. And the TF–IDF is the product of two statistics, term frequency and inverse document frequency.

$$\frac{TF}{IDF} = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \log \frac{|D|}{\left|\left\{j : t_i \in d_j\right\}\right|} \quad (1)$$

Text summarization: Text summarization always tries to determine the central content of documents. And the methods of automatic text summarization are mainly divided into two categories: extractive and abstractive. The extractive summarization is based on the assumption that the core idea of a document can be summarized in one sentence or a few sentences in the document. In this paper, we first preprocess the document and make it a clause. Then the words and sentences are expressed as vectors (word2vec) that the computer can understand. And the sentences are sorted by the following models.

(1) Bag Of Words [22]: The bag of words model defines a word as a dimension, and a sentence is represented as a high-dimensional sparse vector in the space where all words are formed.

(2) Word Embedding [23]: Through word2vec and other techniques, get the low-dimensional semantic vectors of words, sentences, and documents.

3.3. Design Goals. Taking into account the above system model and to solve the problem of neglecting context semantics in the model, the following design goals will be achieved.

(i) Data privacy: our privacy goal is to prevent the cloud learn private information from the outsourced data, the corresponding index, the user queries, and search results.

(ii) Concept attribute access privacy: the cloud cannot know which concept attribute is focused queried and extended.

(iii) Context semantic search: the goal of our scheme is to take context semantic information into consideration in building conceptual graph to achieve more accurate search.

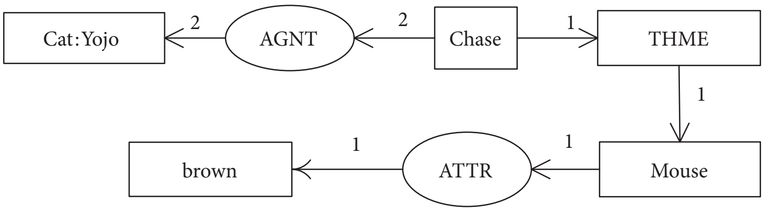


Figure 3: Weighted Conceptual graph.

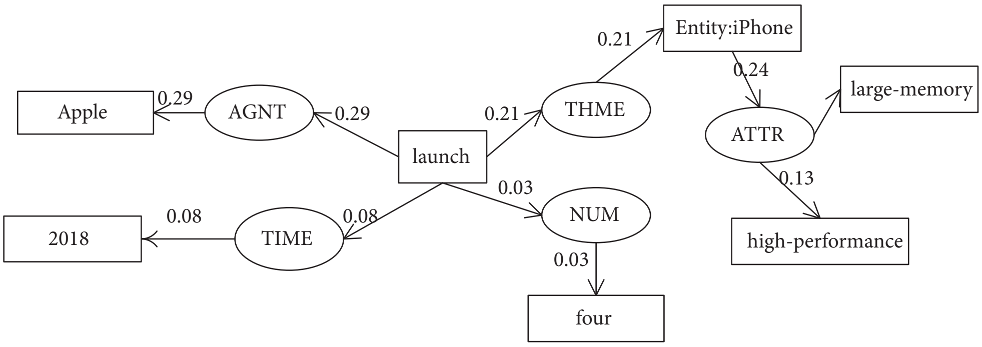


Figure 4: Weighted Conceptual graph for the subject sentence.

4. The Proposed Schemes

The searchable encryption scheme [10] ignores the semantic information of the context during the construction of the index; thus the accuracy of the search matching can be lost. We reconstruct the context-sensitive searchable encryption scheme based on conceptual graphs. First of all, we need to summarize the content of the document. According to the scheme in Section 3.2.2, we can get a topic sentence from the document abstract, and then we establish the corresponding conceptual graph [11].

In our scheme, considering the efficiency of the contextual semantic extension, we only extend the semantic information of the most important topics and construct a semantic network based on conceptual graph of document and then establish a corresponding encrypted index.

4.1. Our Basic Idea. In this section, we will detail our index construction scheme.

4.1.1. Weighted Conceptual Graph. We first introduce the weighted conceptual graph [24], which can help us analyze the importance of each concept attribute in the topic sentence, and reflect the theme of the document. In our scheme, both edges and nodes have weights. And edges' weights are assigned according to the relevance of the semantic flow in the concept relationship as shown in Figure 3.

In our idea, the initial importance of each concept should be the equal. Then we define it as follows.

Definition 1. The more times a concept type appears in a document or more grammatical relations between its conceptual type and other key attributes, the more important it is.

So after we have extracted the central sentence and constructed corresponding conceptual graph, we get all its concept attribute values (rectangular) and we calculate the term frequency (each sentence is considered as a document) and the document frequency of the concept attribute value in its sentence. We use the algorithm to get our weight. We represent the concept value in the concept map as its corresponding weight.

$$q_i = \frac{TF}{IDF\left(ca_i\right)} = DN \times \left(\frac{\log\left(1+tf\right)}{\log\left(df\right)}\right) \qquad (2)$$

Thus, we can effectively obtain topic attributes by statistically weighting concepts, which helps us to extend its context-sensitive semantics. Suppose we obtain the subject sentence of the document: "Apple will launch four high-performance and large-memory iPhones in 2018." Our weighted conceptual graph for the sentence is shown in Figure 4.

4.1.2. Context-Sensitive Expansion of Central Attributes. For the topic sentence of Figure 4, we obtain the theme concept attribute which is "apple," but computer cannot know whether it represents the name of the fruit or the name of the enterprise, which leads to error easily when it is extended semantics and synonyms. So we need to have context-based semantic extensions for the central key attributes.

Our context-sensitive semantic expansion scheme is based on the assumption that the frequent-common attribute in the document has statistical relevance for the same topic. Therefore, we can reflect the connection relation of attributes by statistically analyzing the contextual relationships from the document collection.

We have the following definitions.

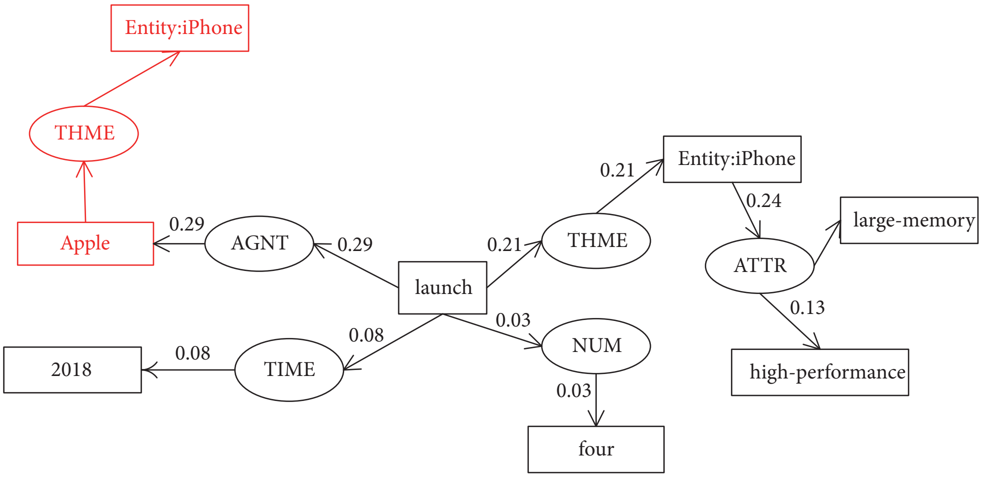


Figure 5: Context-based extension conceptual graph.

Definition 2. The vector of the concept attribute $attr_i$ is represented by

$$attr_i = \left\langle w_{1j}, w_{2j}, \cdots, w_{nj} \right\rangle . \quad (3)$$

n indicates the number of words which are cooccurring with $attr_i$ at least once in the document, and w_{kj} represents the word-to-word weight of word t_k to word $attr_j$.

In our scheme, we define that extended words and key attributes must belong to the same sentence. And it is generally believed that the closer the word to the key attribute in the document, the more times the word appears around the keyword,

Definition 3. Relevance between concept attribute and the word:

$$P\left(attr \mid t_j\right) = w_{kj} \times \frac{e^{-\lambda d(attr,t_j)}}{\sum_j e^{-\lambda d(attr,t_j)}}. \quad (4)$$

λ is the influence factor, and $d(attr, t_j)$ represents the distance between $attr$ and t_j.

When we calculate the relevance of all the extended words, then we need to calculate the relevance of the extended words to the subject sentence.

Definition 4. The relevance of the word t_j to the key sentence A:

$$R\left(t_j, A\right) = \sum_{ai \in A} P\left(t_j \mid a_i\right) \quad (5)$$

Q is a set of all the different concept attributes in the key sentence.

When selecting the extended word, we need to calculate the relevance of the word and the key attributes. At the same time, through Definition 4, we can get an extended attribute which is most relevant to the content of the entire topic sentence. Our scheme extends the semantics based on the context of concept attributes. For example, for Figure 4, we extend its context semantics in Figure 5.

Similarly, for the user's query sentence, we also need to construct a corresponding conceptual graph. And in order to return the search results which best match the user's search intent, our paper adopts the method of [18] to construct a user interest model with semantic information on the user's input topic through the wordnet synset.

4.1.3. Index and Trapdoor Constructions. After we obtain the context conceptual graph, we need to construct corresponding index structure, which can store all semantic information of conceptual graph. We take Figure 5 as an example to illustrate our construction scheme.

First, we design two vectors for the index. The first vector is mainly used to match the semantic structure in the query request. The second vector is used to store the weight of the semantic role, so that we can know the theme of the document. In our scheme, we ignore the conceptual type information in the conceptual graph because it is dispensable in our semantics. Meanwhile, we need to construct a hash table to store the corresponding concept attribute values. For the extended concept attributes, we only need to store it in the corresponding vector, so that the semantic information of the entire conceptual graph can be completely stored through our index structure.

The construction process is as follows:

$$D_1[j] = \begin{cases} \left|c_j\right|, & \left|c_j\right| > 0 \\ 0, & \left|c_j\right| = 0 \end{cases} \quad (6)$$

$$DW_1[j] = \begin{cases} \sum_{|c_j|} q_i, & \left|c_j\right| > 0 \\ 0, & \left|c_j\right| = 0 \end{cases} \quad (7)$$

$$DM_1[j] = \begin{cases} \forall \left(c_{ij}, r_i\right); & \left|c_{ij}\right| > 0 \\ null, & \left|c_{ij}\right| = 0 \end{cases} \quad (8)$$

For the first vector D_1, if the conceptual graph CG contains a semantic role r_i and it has $|c_j|$ number concept

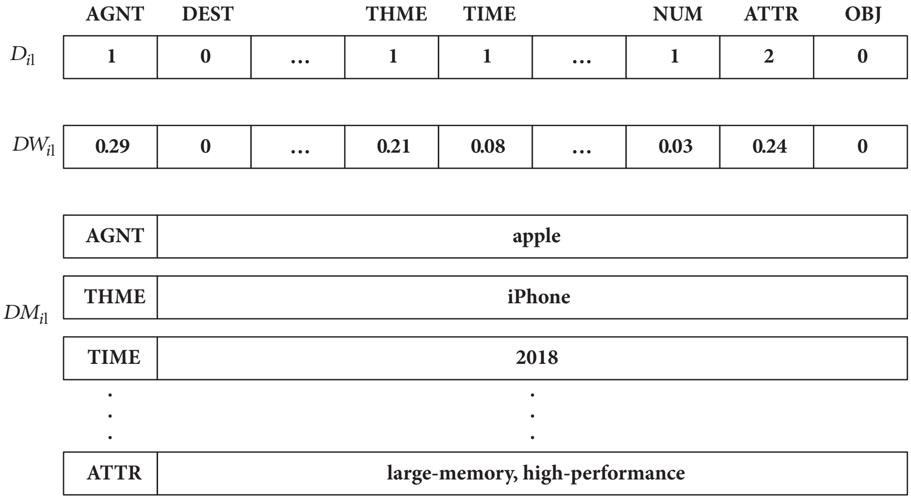


Figure 6: The index structure.

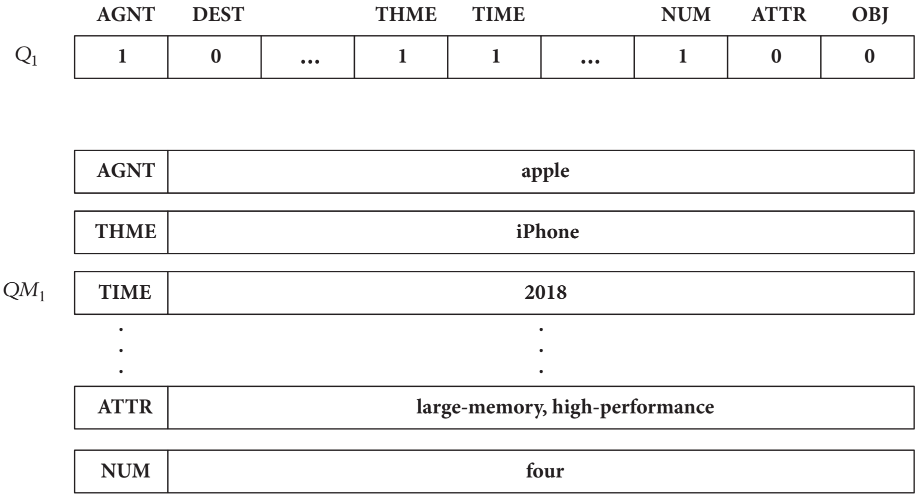


Figure 7: The trapdoor structure.

attributes ($|c_j|$ represents the number of concept attributes), $D_1[j] = |c_j|$. For second vectors DW_1, the weight of each semantic role $DW_1[j]$ is equal to the sum of all the weights of the concept attributes. Meanwhile, we construct the hash table $DM_1[j]$ to store the corresponding concept attribute values. The key is to store the corresponding semantic role; then its value can store its corresponding concept attribute values. The index structure is shown in Figure 6.

Similarly, we can also generate corresponding conceptual graph for user-entered query sentence and also construct corresponding trapdoor structures. For example, the user enters a query statement: "Apple tipped to launch four iPhones in 2018." We get its trapdoor structure as shown in Figure 7.

4.1.4. Retrieval Calculation. Then, we give our retrieval scheme. The data user generates a vectors and hash table Q_1, QM_1 based on the query sentence. The cloud server first calculates the inner product of D_1 vector and Q_1 vector and multiplies the weight vector DW_1 of document semantic role to select the N documents set with the largest correlation score. Then, the cloud server will match DM_1 and QM_1, that is, whether the corresponding semantic roles have corresponding concept attribute values, and calculate the final score so as to filter out the most relevant K documents from the N documents set. As shown in Algorithm 1 *score* is the threshold for $R(D_1, Q_1)$.

4.2. ESSEC Scheme. We use the MRSE framework [10, 25] to construct a searchable encrypted ESSEC scheme based on context-sensitive conceptual graph. At the same time, we combine submatrix techniques to reduce the encryption time in conjunction with the [11] scheme. The encryption scheme contains four steps: KeyGen, BulidIndex, Trapdoor, and Query. By calculating the cosine distance between the two vectors, we can get the similarity score, described as follows.

KeyGen: The data owner first constructs a secret key SK, generating two $(n+2) \times (n+2)$ invertible matrices M_1, M_2

```
Algorithm 1 RCG

Input: $F, D_1, DW_1, DM_1, Q_1, QM_1$
Output: $F(Q)$
For each document $F_i$ in $F$ do
    If $R(D_{i1}, Q_1) = D_{i1} \cdot Q_1 > score$ then
        Calculate $DW_{i1}, DM_{i1}$ and $QM_1$
        $R(DW_{i1}, Q_1) = DW_{i1} \cdot Q_1$
        $R(DM_{i1}, QM_1) = \forall(\exists Val(DM_{i1})\ in\ QM_1)$
        Insert a new element $(R(DW_{i1}, Q_1), R(DM_{i1}, QM_1), FID)$ into $RList$.
    Else return;
    End if
End for
```

Algorithm 1: Retrieval algorithm.

which generated randomly and a $(n+2)$ bit vector S generated randomly, to form $SK = \{S, M_1, M_2\}$.

BulidIndex(D, DM)**:** The scheme generates subindex $\overrightarrow{D}$ by applying dimension expansion and splitting procedures on D, which is similar to the secure KNN algorithm [10]. In this process, we generate two vectors $\{D_i', D_i''\}$. And we set the $(n+1)$-th dimensions in $\overrightarrow{D}$ to a random number ε; $(n+2)$-th dimensions is set to 1. Therefore, $\overrightarrow{D} = (D_i, \varepsilon_i, 1)$. Finally, the encrypted subindex $I_i = \{M_1^T \overrightarrow{D}'_i, M_2^T \overrightarrow{D}''_i\}$ for each encrypted document C_i. And the DM is encrypted by using $\pi(\bullet)$, which is an off-the-self hash function.

Trapdoor(Q, QM)**:** The user generates a n-bit vector $\overrightarrow{Q}$ for query sentence. Then, a similar splitting process will be applied. We extend $\overrightarrow{Q}$ to $(n+1)$-dimension, and $(n+1)$-th is set to 1. Then scale it with a random number $r \neq 0$. And $\overrightarrow{Q}$ is extended to $(n+2)$-dimension. Therefore, $\overrightarrow{Q} = (rQ, r, t)$, t is random. The formulation of T_q is $\{M_1^{-1}\overrightarrow{Q}', M_1^{-1}\overrightarrow{Q}''\}$. Similarly, the QM is encrypted by using $\pi(\bullet)$.

Query: The cloud server calculates the encrypted trapdoor and encrypted index based on cosine measure. The final relevance score is

$$\begin{aligned}
&\cos(I_i, T_Q) \\
&= \cos\left(\left\{M_1^T \overrightarrow{D}'_{i1}, M_2^T \overrightarrow{D}''_{i1}\right\} \cdot \left\{M_1^{-1} \overrightarrow{Q}'_1, M_2^{-1} \overrightarrow{Q}''_1\right\}\right) \\
&= \cos\left(\overrightarrow{D}'_{i1} \cdot \overrightarrow{Q}'_1 + \overrightarrow{D}''_{i1} \cdot \overrightarrow{Q}''_1\right) \\
&= \cos\left\{\left(D_{i1}, \varepsilon, 1\right) \cdot \left(rQ_1, r, t\right)\right\}
\end{aligned} \tag{9}$$

Then cloud server can compare whether $\pi(DM)$ are the same as $\pi(QM_{i1})$ in document set $F(Q_1)$.

5. Performance Analysis

In essence, our proposed scheme is only some post processing further considered compared with the method in [19]. Therefore, the security of our scheme directly inherits the security of the method in [19]. In addition, we adopt the secure KNN inner product scheme [10].

In this section, to assess the feasibility of our scheme, we use java+stanfordNLP to build our experimental platform. Our implementation platform is Windows 7 server with Core CPU 2.85GHz. The dataset is a real-world dataset: CNN set (https://edition.cnn.com/) which is available to construct the outsourced dataset [26]. In our experiment, we use approximately 1000 documents.

5.1. Precision

5.1. Precision. Precision means that users can get what they want based on their queries' sentence. In our scheme, we expand the conceptual graph based on context semantic information. In order to achieve a balance between security and precision, we use 2 layers of index to store all the semantic information of the conceptual graph and also store the extended context semantic information. Thus, the retrieval accuracy of our scheme has a wider range of breadth and deeper precision.

5.2. Efficiency. In our scheme, we need to segment the documents of the dataset and remove the stop-word. We get topic sentences by word2vec, word-embedding, and other NLP methods, but we do not calculate the time, because the time is influenced by the corpus. Thus the time of index construction consists of 2 parts: one is to make a syntactic analysis of the subject sentence and the other is to construct the corresponding index and encrypt the index.

We can see in Figure 9 the relationship between the time of the index construction and the number of documents. Table 1 shows the required time cost and space cost for each index when the size of the document is about 1000. This is because our scheme needs to count the weights of the concept attributes and also needs to extend the context semantics of the central concept attributes. Thus, our index construction needs more time. Our scheme is different from the traditional keywords searchable scheme. We have taken into account the content of the document when constructing the index, which has greatly improved in accuracy and semantic aspects. Meanwhile, compared with the MRSE [10]

Table 1: Index Construction overhead for 1000 documents.

Scheme	Index vectors size	The time of index vectors for each file
MRSE [10]	12898KB	0.9s
USSCG [19]	8394KB	1.79s
Our	10738KB	1.84s

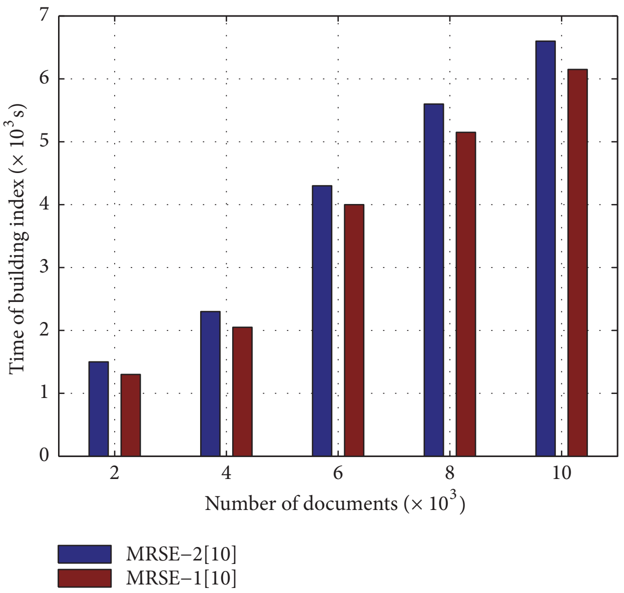


Figure 8: The time cost for generating index vectors in MRSE [10].

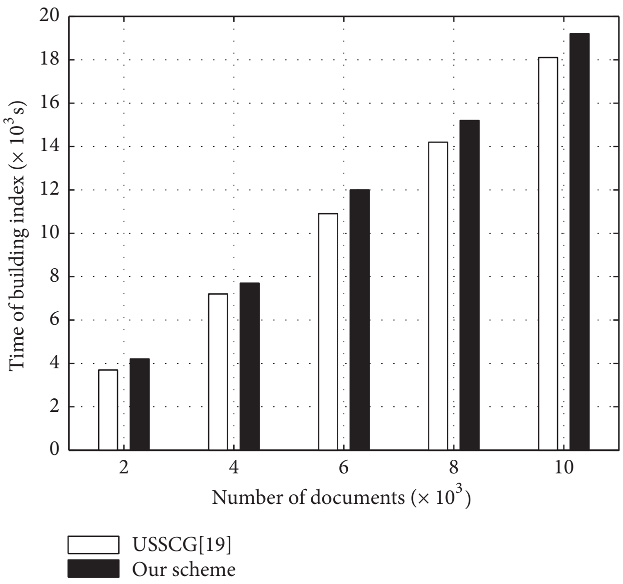


Figure 9: The time cost for generating index vectors.

index construction time (Figure 8), our scheme proved to be acceptable.

Figures 10 and 11 are the analysis diagrams of the relationship between the query time and the number of documents. It can be clearly seen that the query time is proportional to the number of documents, because the increase in the number of documents leads to an increase in the number of conceptual graphs indexes and the increase in the complexity of the context extension so that the query time eventually increases. Despite the results, our scheme has more time than MRSE [10] (Figure 10) and USSCG [19]. However, because our scheme carries the semantic information of the document content, we return more accurate results to compensate for the loss of efficiency.

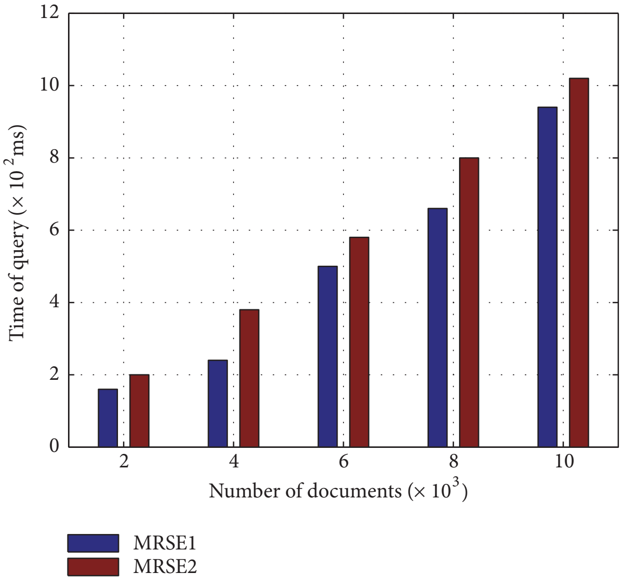


Figure 10: The time cost for query in MRSE [10].

6. Conclusion and Future Work

In this paper, for the first time, we take the relationship between semantic information of the context and conceptual graph into consideration, and we design a semantic search encryption scheme for context-based conceptual graph. By choosing the central key attributes in the topic sentence, not all attributes, our scheme performs a tradeoff between functionality and efficiency. To generate the conceptual graphs, we apply a state-of-the-art technique, i.e., word embedding and Tregex, a tool for simplifying sentences in our method. Also for the literature [10], we put forward a supplementary plan. When constructing the conceptual graph, we considered the semantic information of the context. By extending the context of the central concept attribute, we enhance the relevance of our semantic query and achieve a balance of precision and efficiency. Experimental results demonstrate the efficiency of our proposed scheme.

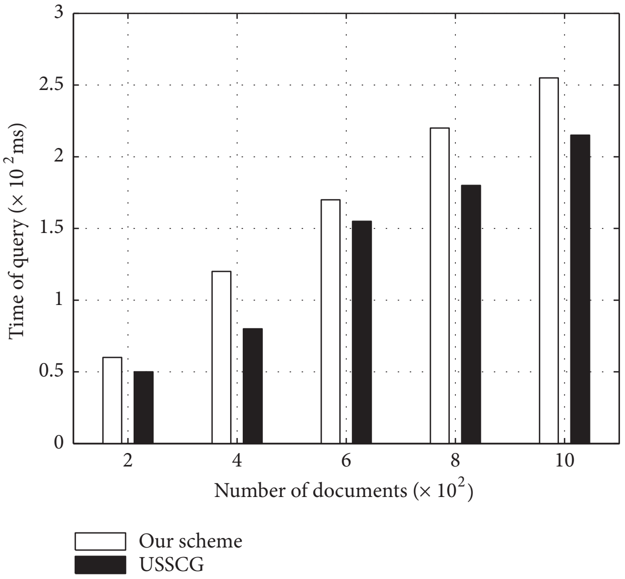


FIGURE 11: The time cost for query.

In the future, we will continue to focus our research on semantic searches using grammatical relations and other natural language processing. In addition, we are considering modifying the process of changing a conceptual graph into a numerical vector which can help improve accuracy and efficiency.

Data Availability

Our dataset is a real-world dataset: CNN set (https://edition.cnn.com/) which is available to construct the outsourced dataset [26]. And we construct the conceptual graphs by [24].

Conflicts of Interest

We declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant U183610040, 61772283, U1536206, U1405254, 61602253, 61672294, 61502242; by the National Key R&D Program of China under grant 2018YFB1003205; by China Postdoctoral Science Foundation (2017M610574); by the Jiangsu Basic Research Programs-Natural Science Foundation under grant numbers BK20150925 and BK20151530; by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund; by the Major Program of the National Social Science Fund of China (17ZDA092), Qing Lan Project, Meteorology Soft Sciences Project; by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

References

[1] J. Leskovec, A. Rajaraman, and J. D. Ullman, "Mining of massive datasets," Cambridge University Press, 2011.

[2] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proceedings of the International Conference on Applied Crygptography and Network Security*, pp. 442–455, 2005.

[3] B. Dan, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *Lecture Notes in Computer Science*, vol. 3027, no. 16, pp. 506–522, 2004.

[4] S. Hensman and J. Dunnion, "Automatically building conceptual graphs using verbnet and wordnet," in *Proceedings of the International Symposium on Information and Communication Technologies*, pp. 115–120, 2004.

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[6] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy, 2000*, pp. 44–55, IEEE, Berkeley, Calif, USA, May 2000.

[7] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," *Lecture Notes in Computer Science*, pp. 31–45, 2004.

[8] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," *ICICS*, pp. 414–426, 2005.

[9] Y. L. Liu, H. Peng, and J. Wang, "Verifiable Diversity Ranking Search Over Encrypted Outsourced Data," *Computers Materials & Continua*, vol. 55, no. 1, pp. 37–57, 2018.

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

[11] W. Sun, B. Wang, N. Cao et al., "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013*, pp. 71–81, May 2013.

[12] 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 Generation Computer Systems*, vol. 30, no. 1, pp. 179–190, 2014.

[13] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[14] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proceedings of the IEEE INFOCOM, 2010*, pp. 1–5, IEEE, San Diego, CA, USA, 2010.

[15] Z. Fu, J. Shu, X. Sun, and N. Linge, "Smart cloud search services: Verifiable keyword-based semantic search over encrypted cloud data," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 762–770, 2014.

[16] Z. Fu, X. Wu, Q. Wang, and K. Ren, "Enabling central keyword based semantic extension search over encrypted outsourced data," *IEEE Transactions on Information Forensics & Security*, vol. 12, no. 12, 2017.

[17] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 2112–2120, May 2014.

[18] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel & Distributed Systems*, vol. 27, no. 9, p. 1, 2016.

[19] Z. Fu, F. Huang, X. Sun, A. V. Vasilakos, and C. Yang, "Enabling semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Transactions on Services Computing*, 2016.

[20] J. F. Sowa, "Conceptual structures: information processing in mind and machine," 1983.

[21] G. W. Mineau, B. Moulin, and J. F. Sowa, *Conceptual Graphs for Knowledge Representation*, Springer Berlin Heidelberg, 1993.

[22] R. Ihaka and R. Gentleman, "R: a language for data analysis and graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

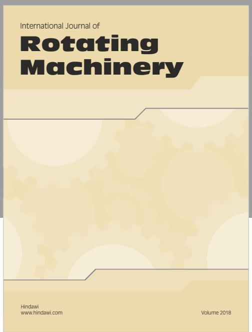
[24] S. Miranda-Jiménez, A. Gelbukh, and G. Sidorov, "Summarizing Conceptual Graphs for Automatic Summarization Task," in *Proceedings of the International Conference on Conceptual Structures*, vol. 7735, pp. 245–253, 2013.

[25] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems, ICDCS 2010*, pp. 253–262, June 2010.

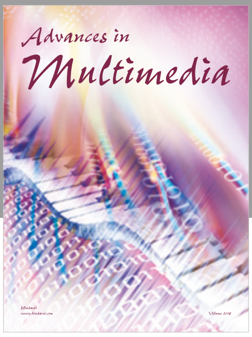
[26] Y. L. Liu, N. Hu, H. Xu, H. Ning, and L. K. Wu, "A real-time monitoring technique for local plasticity in metals based on Lamb waves and a directional actuator/sensor set," *Computers, Materials & Continua*, vol. 40, no. 1, pp. 1–20, 2014.

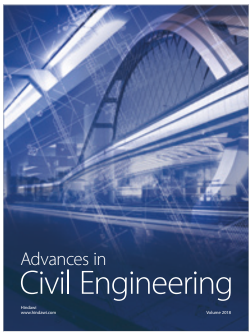
Journal of
Engineering

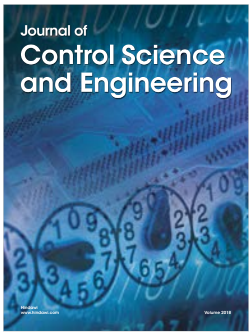
The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

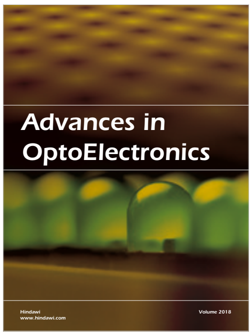

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

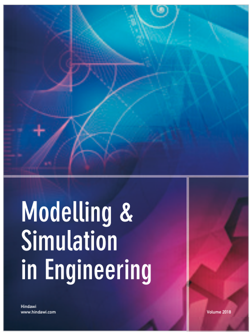
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

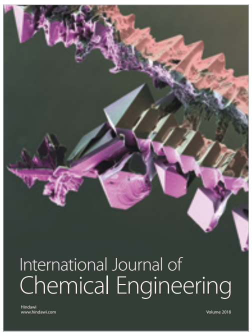

Advances in
OptoElectronics

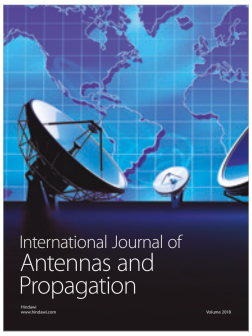
VLSI Design

International Journal of
Navigation and
Observation

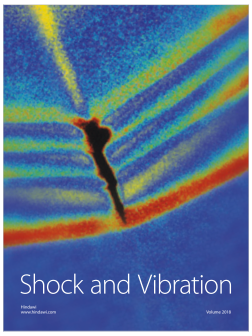

Modelling &
Simulation
in Engineering

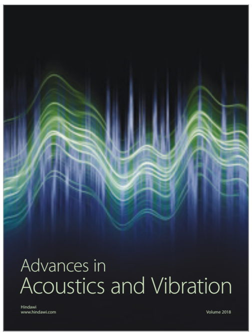
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration