Compound Keyword Search of Encrypted Cloud Data by using Semantic Scheme

Harsh Gupta[1], Kartik Ahirrao[2], Noopur Sonaje[3], S.N. More[4]

[1,2,3]Dept. of Computer Engineering, LOGMIEER College, Maharashtra, INDIA
[4]Professor S.N. More, Dept. of Computer Engineering, LOGMIEER College, Maharashtra, INDIA

---***---

Abstract - *Keyword search over encrypted data is crucial for accessing the outsourced sensitive data in cloud computing. In some situations, the keywords which the user searches on are only semantically related to the data rather than via an exact or fuzzy match. Therefore, semantic-based keyword search over encrypted cloud data becomes of foremost importance. However, existing schemes usually depend upon a global dictionary, which not only affects the accuracy of search results but also causes in ability in data updating. Additionally, although compound keyword search is frequent in practice, the existing approaches only process them as single words, which split the original semantics and achieve low accuracy. To address these limitations, we initially posit a compound concept semantic similarity (CCSS) calculation method to compute and measure the semantic similarity between compound concepts. Next, by combining CCSS with Locality-Sensitive Hashing function and the secure k-Nearest Neighbor scheme, a semantic-based compound keyword search (SCKS) scheme is proposed. SCKS achieves not only semantic-based search but also multi-keyword search and ranked keyword search all together. Additionally, SCKS also eliminates the predefined global library. It can also efficiently support data update. The experimental results on real-world dataset indicate that SCKS introduces low overhead on computation and the search accuracy outperforms the existing schemes.*

Key Words: **Searchable encryption, Semantic-based keyword search.**

1. INTRODUCTION:

In cloud computing, an exceeding number of personal or enterprise users outsource their data to cloud storage to enjoy the benefits of ”pay-on-demand” services and high computation execution. To preserve privacy, users choose to encrypt the data before outsourcing. Thus, the traditional keyword search cannot be directly executed on the encrypted data, which restrains the utilization of data. Semantic-based keyword search not only is convenient for users but also exactly expresses users' intentions. Specifically, in some circumstances, users might not be familiar with the encrypted documents stored in cloud storage or might only want the semantically related results; therefore, the search keywords are usually semantically interconnected to the document rather than through an exact or even a fuzzy match. For example, the predefined keyword of a document is “cloud-based storage”, and the keyword that a user searches is “distributed storage”. Obviously, these two considered words are neither an exact nor a fuzzy match, but they are semantically related. Hence, the semantic-based keyword search is of practical importance and has attracted much attention. However, the existing approaches must rely on a predefined global dictionary whose quality greatly influences the accuracy of the search result. Moreover, when the dataset is being outsourced to the cloud, update operations that include inserting new documents and modifying and deleting existing documents are frequent. Because the predefined dictionary is constructed based on all documents in the dataset, the update of a single document can cause the reconstruction of the dictionary and even all document indexes, which is inefficient. The semantic similarity between keywords in a query and keywords of documents is important because it also determines the exactness of search results. However, in the aforementioned approaches, the semantic information used for measuring the semantic sameness was mined from some knowledge bases (KBs) (such as corpus and thesaurus) containing noise data, which cause the semantic similarity to be inaccurate. Compared with other KBs, ontology has good support for logic reasoning and can structurally express the semantic information of concepts. Several ontology-based viewpoints have been proposed to assess the similarity between concepts through mining ontology information from different aspects. Each document generally consists of more than one keyword, the index of a document is associated with multiple keyword vectors. Locality-Sensitive Hashing (LSH) function is able to hash similar items to the similar bucket with elevated probability. Hence, we construct the document index by using LSH to map multiple keyword vectors into only one vector. The frequency of the keyword in the document is also considered and is positioned into the index vector as the value of corresponding element. A semantic-based keyword search scheme returns results according to the semantic Interconnectedness between documents and a query.

2. History and Background:

Owners prefer to outsource documents in an encrypted form for the purpose of privacy preserving. Therefore, it is crucial to develop structured and reliable cipher text search techniques.

One challenge is that the connection between documents will be normally secreted in the process of encryption, which will ultimately lead to significant search accuracy performance degradation. Also the quantity of data in data centers has encountered a dramatic growth. This will make it even more challenging to design cipher text search schemes that can provide efficient and reliable online information retrieval on large volume of encrypted data. A conventional way to reduce information leakage is data encryption.

However, this will also make server-side data utilization, such as discerning on encrypted data, become a very demanding task. In the recent years, many cipher text search schemes have been proposed by researchers by incorporating the cryptography techniques. These schemas have been proven with provable security, but their methods need massive operations. They also have high time complexity.

Therefore, the traditional methods are not acceptable for the big data scenario where data quantity is very big and applications require online data processing. Sanctioning keyword search directly over encrypted data is a desirable technique for effective utilization of encrypted data outsourced to the cloud. Existing solutions provide multi keyword exact search that does not indulge keyword spelling error, or single keyword fuzzy search that indulges typos to a certain extent [1].

The current fuzzy search schemes are dependent upon building an expanded index that covers possible keyword misspelling, which lead to significantly larger index file size and higher search complexity. In cloud computing, scalable and elastic storage and computation resources are provisioned as measured services through the Internet. Outsourcing data services to the cloud allows organizations to enjoy not only monetary savings, but also simplified local IT management since cloud infrastructures are physically hosted and maintained by the cloud providers [2].

To minimize the risk of data leakage to the cloud service providers, data owners prefer to encrypt their sensitive data, e.g., user records, financial transactions, before outsourcing to the cloud, while holding on to the decryption keys to themselves and other authorized users. This in turn supports data utilization a challenging problem. Due to the alluring features of cloud computing, huge amount of data have been stored in the cloud.

Even though loud based services offer many upper hand advantages, privacy and security of the sensitive data is of a big concern. To reduce the concerns, it is desirable to outsource sensitive data in encrypted form. Encrypted storage protects the data against illegal access, but it complicates some basic, yet important functionality such as the search on the data.

However, all of them handle exact query matching rather than similarity matching; a necessary requirement for real world applications. Even though some sophisticated secure multi-party computation based cryptographic techniques are available for comparability tests, they are computationally intensive and do not scale for large data sources[3].

As the data produced by enterprises that need to be stored and utilized are rapidly increasing at an increasing rate, data owners are usually inspired to outsource their local complex data management systems into the cloud for its great workability and economic savings. As sensitive cloud data may have to be encrypted before outsourcing, which obsoletes the traditional data utilization service based on plaintext keyword search, how to enable privacy-assured utilization mechanisms for outsourced cloud data is thus of paramount importance.

Considering the huge number of on-demand data users and vast amount of outsourced data files in cloud, the problem is particularly demanding, as it is extremely difficult to meet also the general and practical requirements of performance, system usability. Aside from abolishing the local storage management, storing data into the cloud serves no purpose unless and until they can be easily searched and utilized.

Thus, exploring privacy-assured and effective search service over encrypted cloud data is of paramount importance [4]. Data owners get inspired to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data have to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance.

Considering the huge amount of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Related works on searchable encryption focus on single keyword search. It is also known as Boolean keyword search, and it rarely sorts the search results.

To meet the effective data retrieval need, the large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results. These ranked search system enables the data users to find the most similar information quickly, rather than burdensomely sorting through every match in the content collection making our process exhaustive.

Ranked search can also elegantly eliminate the unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the “pay-as-you-use” cloud paradigm[5]. Cloud computing infrastructure is a promising and a new technology and greatly accelerates the development of large scale data storage, processing and distribution.

However, security and privacy becomes a major issue when data owners outsource their sensitive data onto the public cloud servers that are not within their trustworthy management domains. To avoid information this leakage, private data has to be encrypted before uploading it onto the cloud servers, which makes it a big challenge to support efficient keyword based queries and rank the matching results on the encrypted data. A huge amount of current works only consider a single keyword queries without appropriate ranking schemes.

In the existing multi-keyword ranked search approach, the keyword dictionary is static and cannot be expanded that easily when the number of keywords also increases. Flexible and economic strategy for data management and resource sharing is provided by cloud computing infrastructures. It can reduce the hardware and software costs and also system maintenance overheads.

It can also provide a convenient communication medium to share the resources across data owners and data consumers. For example Amazon Web Services [6].

3. System Architecture:

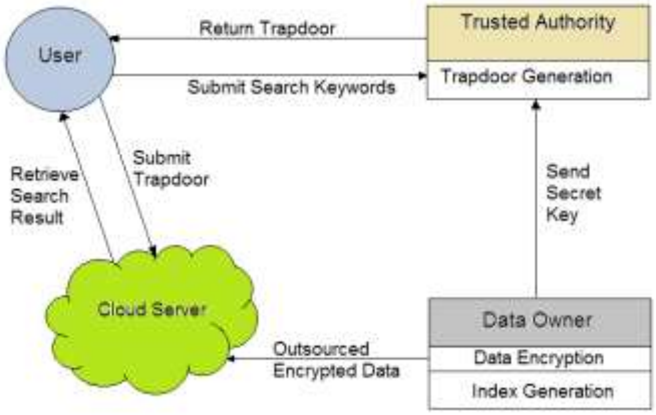

Fig 1: The model of keyword search

A document index is denoted by a vector generated with the keywords of the document, and a secure index is the encrypted index. A query is a vector generated with the keywords of a search, and a trapdoor is a query which is encrypted. In general, the document can be encrypted by traditional encryption schemes such as AES.

- **Keygen : -** Execution is done by the data owner or a trusted authority (TA). Taking a security parameter d as input, it outputs a system symmetric key.
- **BuildIndex: -** It is executed by the data owner. Based on the symmetric key sk and the document D, the algorithm generates the secure index.
- **Trapdoor: -** This is performed by the data owner. With the keyword set Q that the user wants to search, the algorithm generates the corresponding trapdoor.
- **Search: -** It is performed by the cloud server. Based on the trapdoor and each secure index stored in the server, the cloud server determines the correlation coefficients between the query Q and each document D and returns the ranked correlation coefficients to the user.

The data owner publishes the encrypted documents. He also secures the indexes to the cloud server. To reduce the computation burden, the data owner is allowed to outsource the generation of the trapdoor to TA by giving the private key to it. In this case, whenever a user wants to search over the encrypted documents, he/she submits the keywords to TA which generates the corresponding trapdoor and returns it to the user. Then, the user sends the trapdoor to this cloud server. Finally, the cloud server executes the search algorithm with the trapdoor on all secure indexes and returns the relevant documents to the user. TA is usually an internal server. If there is no such trustable server in the system, the trapdoor can be initiated by the data owner. Also, some existing schemes assume that the authorization between the data owner and users is carried out effectively.

1. **Locality-Sensitive Hashing function: -**The Locality-sensitive hashing (LSH) function hashes input items so that these common items map into the similar “bucket” with high probability. It is able to diminish the dimensionality of high-dimensional data.

2. **Secure k-Nearest Neighbor encryption: -** It is the fundamental query operations in some applications. The goal of SkNN is to firmly spot the k-nearest points in the encrypted database to a given encrypted query, without allowing the data server to obtain the content contained in the database or the query.

3. **Compound Concept Semantic Similarity Calculation Method: -** Depending upon their semantic constituents, the compound concepts can be divided into two types: Endocentric structure and Exocentric structure. The endocentric structure is true when one or more constituents of a compound can play the central role and serve as a definable subject heading.

4. **Semantic Similarity Calculation:-** To measure the semantic sameness of compound concepts, we put forth a novel approach that considers the concept constituent features and several other factors influencing similarity.

5. **Semantic Compound Keyword Based Search Scheme: -** The topic set in a field is used to construct the semantic vector for each keyword. More specifically, in the keyword vector, each element correlates to a field topic, and its value is the similarity between the topic and the keyword, which is obtained using CCSS. Because the topics are almost invariable, the dimensionality of the keyword vector will not change with the adding or deleting of the keywords or documents, which is helpful in supporting data update.

6. **Security Enhanced SCKS: -** The nemesis is allowed to submit queries adaptively, i.e. submitting the next query after receiving the outcomes of previous queries. Thus, the adversary can decide the next query depending upon the previous outcomes.

4. CONCLUSION:

To accurately extract the semantic information of keywords, we first propose an ontology-based compound concept semantic similarity calculation method (CCSS), which greatly improves the precision of sameness measurement between compound concepts by comprehensively considering the compound features and a variety of information sources in ontology. Low overhead on computation and that the search precision outperforms the current schemes.

REFERENCES:

[1]C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, and A. Y. Zomaya, “An efficient privacy- reserving ranked keyword search method,”IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 4,pp. 951–963, 2016.

[2]B. Wang, S. Yu, W. Lou, and Y. T. Hou, “Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud,” in IEEE International Conference on Computer Communications, 2014, pp. 2112–2120.

[3]M. Kuzu, M. S. Islam, and M. Kantarcioglu, “Efficient similarity search over encrypted data,” in IEEE 28th International Conference on Data Engineering (ICDE), 2012, pp. 1156–1167.

[4]C. Wang, K. Ren, S. Yu, and K. M. R. Urs, “Achieving usable and privacy-assured similarity search over outsourced cloud data,” in 2012 Proceedings of IEEE INFOCOM, 2012, pp. 451–459.

[5]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.

[6]R. Li, Z. Xu, W. Kang, K. C. Yow, and C. Z. Xu, “Efficient multikeyword ranked query over encrypted data in cloud computing,” Future Generation Computer Systems, vol. 30, no. 1, pp. 179–190,2014.