ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Privacy Preserving Multi-keywordSearch Over Encrypted CloudData

Vidhu Edavana

M Tech P.G. Scholar, Department of CSE, MIT Anjarakandy, Kannur, Kerala, India

ABSTRACT: The amount of data generated by individuals and enterprises is rapidly increasing.With the emerging cloud computing paradigm, the data and corresponding complex management tasks can be outsourced to the cloud for the management flexibility and cost savings. Unfortunately, as the data could be sensitive, the direct data outsourcing would have the problem of privacy leakage. The encryption can be used, before the data outsourcing, with the concern that the operations can still be accomplished by the cloud. We consider the multi-keyword similarity search over outsourced cloud data. In particular, with the consideration of the text data only, multiple keywords are specified by the user. The cloud returns the files containing more than a threshold number of input keywords or similar keywords, where the similarity here is defined according to the edit distance metric.

We propose five solutions, where file access privacy provides the owner's private file access details, and keyword access privacy shares keyword access details to the owner, and a novel use of hash table's bit pattern provides the speedup of search task at the cloud side and help to remove unused keywords. Alphanumeric keyword similarity search is another main advantage of this paper. If any document with exact keyword matching condition blocks unauthorized access with similarity search. Our final design to achieve the search is secure against insider threats and efficient in terms of the search time at the cloud side.

KEYWORDS:Cloud Computing, Privacy, Similarity Search, Multi-keyword Search, Bloom Filter.

I. Introduction

Cloud computing has become a new computing paradigm. Currently, an increasing number of individuals and enterprises are generating a huge amount of data everyday. It is no longer economically feasible to maintain their own hardware and staffs for data management. Recently, a reasonable and popular choice to mitigate the burden of data management is to outsource the complex data management task to the cloud with the major benefit of cost savings. One may have concern that the cloud cannot always be trusted; it may purposely and unsolicitedly examine the outsourced data [2].

To keep the advantage of cost savings and protect the data privacy, the data, before outsourced to the cloud, need to be encrypted. Despite the success in gaining the data privacy, data encryption does not allow the cloud to answer the user's queries on the data. A straightforward solution for the user to overcome such a difficulty is to simply download the entire data set. This, however, is practically unfeasible because of the huge volume of the incurred bandwidth consumption.

The problem of retrieving information from the encrypted files has already been very challenging. In the context of outsourced cloud data, the problem is even aggravated by a large number of on-demand users and a huge amount of outsourced data files. Therefore, with the given considerations, it is extremely difficult to meet the requirements of both retrieval feasibility and system performance.

In the current use, text data that can be seen everywhere would be the one delivering the majority of information. An important method of retrieving information on the text data is the keyword search, in which only the text files containing the specific keywords are returned to the user. However, possibly due to the lack of the files perfectly matching the input keywords, the keyword search may return an empty result. In this sense, the user naturally turns to seek for the similar result. Here, the similar result could be the files containing part of input keywords or containing the words similar to the input keywords. Such similar keyword search can find numerous applications, such as record linkage [3] and biological database [6]. Due to its ability in enhancing system usability and overall user experience, the research on the similar keyword search has been conducted extensively.

DOI: 10.15680/IJIRCCE.2015. 0312018

II. RELATED WORK

The basic idea behind the very first searchable encryption is to encrypt each word in a text file individually. Then, as can easily be known, the search cost of the given basic searchable encryption would be very high. The subsequent research efforts are put to develop an index that can support more efficient keyword search. Another line of research on searchable encryption is to enrich search predicates; therefore, conjunctive keyword search, subset query, and range query over encrypted data are also introduced. The advantage of searchable encryption is its provable security. Nonetheless, in the setting of secure ranked search, only the number of keyword matches is concerned, and the similarity between the input keywords and the actual words in text is not taken into account. On a different front, privacy assured similarity search, where the files containing exactly the same keyword or containing similar keyword are returned, has also been studied. However, in the setting of privacy assured similarity research, only single keyword is allowed, restricting the practical use.

Edit distance is a quantitative metric to measure the similarity between two strings [4]. The edit distance $ed(w_1, w_2)$ between two words is defined as the minimum number of operations required to transform from one word to another. The operations considered here are character insertion, deletion, and replacement.

One of the characteristics of the Bloom filter is that the query result is always correct if the content to be queried is indeed stored in the Bloom filter. a Bloom filter [1] consists of an array of b bits. Together with k independently and randomly selected hash functions, $h_1....h_k$, with range [0,b_1], it is used to represent a set of elements with the support of membership query. Assume that a Bloom filter B is used to represent a set $S = s_1,...,s_m$ of m elements. To insert an element s_i , the bits $B[h_j(s_i)]$ for $1 \le j \le k$ are set to 1. The bit remains unchanged when being already set to 1. To check whether an element x is in the set S, we can check whether the bits $B[h_j(x)]$ for $1 \le j \le k$ are all 1â€™s. If and only if they are all equal to 1, x is deemed to be an element of S. The size b of the Bloom filter is independent of the size of elements and can be constant, which is very memory efficient. Nevertheless, the membership query on the Bloom filter has false positive but has no false negative.

The problem of multikeyword similarity search over outsourced cloud data. Given a collection $C = f_1....f_n$ of encrypted files, a set $W = w_1...w_p$ of predefined distinct keywords, a set $X = x_1....x_q$ of keywords, a threshold d for the minimum edit distance, and a threshold Î´ for the minimum occurrences of keywords appearing in the file, the result of the multikeyword similarity search is C_X such that, for each file $f \in C_X$, $\sum_{i=1}^{q} \beta_i \ge \delta$, where β_i is defined as $\beta_i = | S_{k(f),d} \bigcap S_{xi,d} |$ with K(f) = $\bigcup_w S_{w,d}$, where wâ€™s are the keywords extracted from the file f , denoting the set of all keyword variants contained in the file f .

III. PROPOSED SYSTEM

A. *Methodology:*

While reading [1],we happened to identify some issues as described below. The issues related to

1. File access privacy.
2. Keyword access privacy.
3. Use of Bloom Filter.
4. Security while similarity search.

B. *Problem Definition:*

This section discusses about some of the main issues of [1]. They are:

- File Access Privacy

  Since as per [1], both the owner and cloud could not know which file is accessed more frequently.The owner is unaware of his files,which are accessed or not.So they cannot identify and remove unused old files.Removal of unused file may save more memory space.

- Keyword access privacy

As in file access privacy,keyword access privacy also restrict owner to understand whether the keywords are frequently accessed or not. So they cannot identify and remove unused keywords. Increased unused keywords increase search complexity and hence decrease performance.

- Use of Bloom Filter
  Bloom filter is used to store and filter search keywords in [1].Since it does not support delete operation, owner cannot remove keywords from the list. So unused keywords cannot be deleted. Increased unused keywords increase search complexity and hence decrease performance.
- Security while similarity search
  Since it uses similarity search for accessing document,and edit distance used for it is common for all document, there is a chance for retrieval of unauthorized document. Consider an example. If a bank is a cloud owner and they put their user details in cloud, if an account holder searched for his account,he may get other holder's document,since edit distance for bank account is very small.

C. *Proposed System:*

This section consists of the solutions for the above mentioned problems

- File Access Privacy
  To solve this problem we need to store document accessing details in the cloud server, which must be accessible only by the corresponding owners of the document only. For the efficient storing of document accessing details we construct a vector D_v :(document vector) with fields $< doc\_id, user\_id, count >$ .By analyzing this document vector D_v value, the owner can identify and remove the least used or unused old documents from his account.
- Keyword access privacy
  To solve this problem we need to store keyword accessing details in the cloud server, which must be accessible only by the corresponding owners of the document only. For the efficient storing of keyword accessing details we construct a vector K_v :(Keyword vector) with fields $< keyword, doc\_id, count >$ .By analyzing this keyword vector K_v value, the owner can identify and remove the least used or unused old keyword document pair from his account.
- Use of Bloom Filter
  Bloom filter does not allow us to remove or replace any keyword.So instead of using bloom filter here new proposal is hash table.So owners can add or remove their keyword document pair when they would like.And hence we can improve performance.
- Security while similarity search
  We can eliminate this issue by using a separate list other than inverted list[1], the list can be named as equal list and the list contains 2 fields like $< keyword, doc\_id >$ and contents of the list is keywords without edit distance or edit distance [3] equal to 0.And this list can be stored by using the method called 'Homomorphic Encryption'[5].

IV. Expected Results

The issues solved when applying

1. File access privacy.
   This feature helps data owner to identify and remove unused files and hence avoid unwanted memory usage
2. Keyword access privacy.
   This feature helps document owner to identify and remove unused keywords and improve search efficiency
3. Use of Bloom Filter.
   This feature helps owner to remove keywords from bloom filter and with this they can reduce false positive outputs. Which helps to reduce search time.
4. Security while similarity search.
   This concept protect data from unauthorized retrieval due to similarity search.

V. Conclusion

We identified five problems in the existing system. File Access Privacy, both the owner and cloud could not know which file is accessed more frequently. Keyword access privacy, keyword access privacy also restrict owner to

DOI: 10.15680/IJIRCCE.2015. 0312018

understand whether the keywords are frequently accessed or not. Use of Bloom Filter, owner cannot remove keywords from the list. Security while similarity search, there is a chance for retrieval of unauthorized document.

Here we proposed some solutions for the above mentioned problems. And by implementing the proposed solution we can improve efficiency and performance of search process.

REFERENCES

1. Yu, C.M., Chen, C.Y., and Chao, H.C., *Privacy-preserving multikeyword similarity search over outsourced cloud data*, Systems Journal, IEEE pp no. 99, 1–10,2015.
2. Liu,F.H., Lo, H.-F., Chen, L.-C., and Lee, W.-T., *Comprehensive security integrated model and ontology within cloud computing*, Journal of Internet Technology **14**, no. 6, 935–946, 2013.
3. Wang,D., Fu, S., and Xu, M.,*A privacy-preserving fuzzy keyword search scheme over encrypted cloud data*, Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on, vol. 1, pp. 663–670, Dec 2013.
4. Atallah, M.J., *Algorithms and theory of computation handbook*, CRC press, 2002.
5. Yu,j., Lu, P., Zhu, Y.,Xue, G., and Li, M.,*Toward secure multikeyword top-k retrieval over encrypted cloud data*, Dependable and Secure Computing, IEEE Transactions on **10**, no. 4, 239–250, 2013.
6. Zhang, Z.,Hadjieleftheriou, M.,Ooi,B.C., and Srivastava, D.,*Bed-tree: an all-purpose index structure for string similarity search based on edit distance*, Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, pp. 915–926, 2010.

BIOGRAPHY

Vidhu Edavanais anM Tech PG Scholar in Department of Computer Science and Engineering, Malabar Institute of Technology, Anjarakandy, Kannur, Kannur University. He worked with Quest Innovative Solutions Pvt Ltd, as a Software Developer. His research interest is in Cloud Computing.