Identifying Keyword Search Using GDFS

Dr. K.L Shunmuganathan[1], Dr. K. Nattar Kannan [2], N.A Lawrance [3]
1-Principal, 2-Professor & HoD, 3- PG Student
Department of Computer science and Engineering
Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India
principal@dce.edu.in, nattarkannank.csc@dce.in, lawra1983@gmail.com

Abstract*:* A secure Multi-Keyword ranked search scheme over cloud data is presented; it supports dynamic update operation like deletion and insertion. Greedy Depth first search algorithm is used to provide efficient multi keyword ranked search. The scheme aims to achieve sub linear search efficiency by exploring a special tree based index and an efficient search algorithm. A secure tree-based search scheme over the cloud data, It supports multi=keyword ranked search and dynamic operation on the document collection. The secure KNN algorithm is utilized to index and query vectors and ensure accurate relevance score calculation between index and query vectors. A searchable scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.

Keywords: Security, k-NN classifier, Outsourced databases, Encryption.

I. Introduction

Recently, the cloud computing paradigm is revolutionizing the organizations'[1] way of operating their data particularly in the way they store, access and process data. As an emerging computing paradigm, cloud computing attracts many organizations to consider seriously regarding cloud potential in terms of its cost efficiency, flexibility, and offload of administrative overhead. Most often, organizations delegate their computational operations in addition to their data to the cloud [2][3]. Despite tremendous advantages that the cloud offers, privacy and security issues in the cloud are preventing companies to utilize those advantages. When data are highly sensitive, the data need to be encrypted before outsourcing to the cloud. However, when data is encrypted, [4] irrespective of the underlying encryption scheme, performing any data mining tasks becomes very challenging without ever decrypting the data. There are other privacy concerns, demonstrated by the following example [5][6]. Suppose an insurance company outsourced its encrypted customers database and relevant data mining tasks to a cloud. When an agent from the company wants to determine the risk level of a potential new customer, the agent can use a classification method to determine the risk level of the customer. First, the agent needs to generate a data record q [7] for the customer containing certain personal information of the customer, e.g., credit score, age, marital status, etc. Then this record can be sent to the cloud, and the cloud will compute the class label for q. Nevertheless, since q contains sensitive information, to protect the customer's privacy, q [8] should be encrypted before sending it to the cloud. The above example shows that data mining over encrypted data (denoted by DMED) on a cloud also needs to protect a user's record when the record is a part f a data mining process[9][10]. Moreover, cloud can also derive useful and sensitive information about the actual data items by observing the data access patterns even if the data are encrypted . Ranked search can also gracefully remove redundant network traffic by transferring the most relevant data, which is highly attractive in the "pay-as-you-use" cloud concept. For privacy protection, such ranking operation on the other hand, should not reveal any keyword to related information. To get better the search result exactness as well as to improve the user searching experience, it is also essential for such ranking system to support multiple keywords search, as single keyword search often give up far too common results.

II. Implementation

2.1 EXISTING SYSTEM

A general approach to protect the data confidentiality is to encrypt the data before outsourcing.

Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute keyword search over cipher text domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword Boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi-keyword ranked search achieves more and more attention for its practical applicability. Recently, some *dynamic* schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server.

2.1.1DISADVANTAGES OF EXISTING SYSTEM:

Huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical.

Existing System methods not practical due to their high computational overhead for both the cloud sever and user.

2.2 PROPOSED SYSTEM:

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used “term frequency (TF) × inverse document frequency (IDF)” model are combined in the index construction and query generation to provide multi-keyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a “Greedy Depth-first Search” algorithm based on this index tree.

The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors.

To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known cipher text model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model.[11]

2.2.1 ADVANTAGES OF PROPOSED SYSTEM:

Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents.

We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.

Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our “Greedy Depth-first Search” algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

2.2.2 The K-NN methodology:

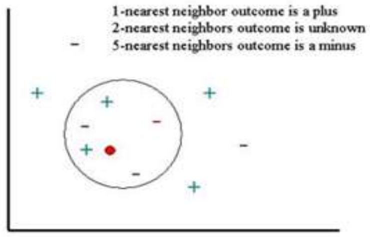
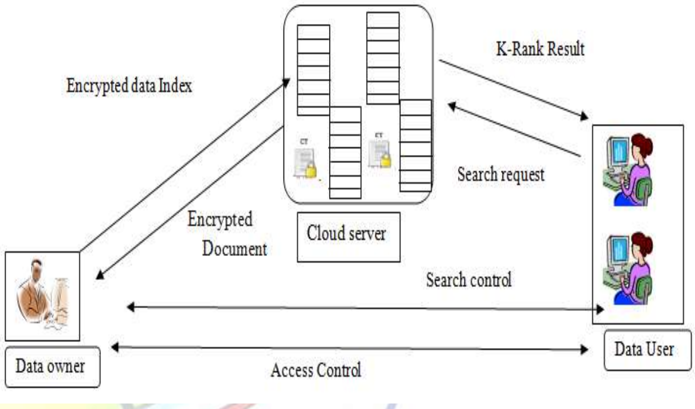

1-nearest neighbor outcome is a plus
2-nearest neighbors outcome is unknown
5-nearest neighbors outcome is a minus
+
-

Fig 3.1 K-NN Implementation.

SYSTEM ARCHITECTURE



III.LIST OF MODULES

- Data Owner Module
- Data User Module
- Cloud server and Encryption Module
- Rank Search Module

3.1Data Owner Module

This module helps the owner to register those details and also include login details. This module helps the owner to upload his file with encryption using RSA algorithm. This ensures the files to be protected from unauthorized user. Data owner has a collection of documents *F ={f*1; *f*2; *:::; fn}* that he wants to outsource to the cloud server in encrypted form while still keeping the capability to search on them for effective utilization. In our scheme, the data owner firstly builds a secure searchable tree index *I* from document collection *F*, and then generates an encrypted document collection *C* for *F*. Afterwards, the data owner outsources the encrypted collection *C* and the secure index *I* to the cloud server, and securely distributes the key information of trapdoor generation and document decryption to the authorized data users. Besides, the data owner is responsible for the update operation of his documents stored in the cloud server. While updating, the data owner generates the update information locally and sends it to the server.

3.2 Data User Module

This module includes the user registration login details. This module is used to help the client to search the file using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required file and register the user details and get activation code in mail email before enter the activation code. After user can download the Zip file and extract that file. Data users are authorized ones to access the documents of data owner. With *t* query keywords, the authorized user can generate a trapdoor *TD* according to search control mechanisms to fetch *k* encrypted documents from cloud server. Then, the data user can decrypt the documents with the shared secret key.

3.3 Cloud Server and Encryption Module:

This module is used to help the server to encrypt the document using RSA Algorithm and to convert the encrypted document to the Zip file with activation code and then activation code send to the user for download. Cloud server stores the encrypted document collection *C* and the encrypted searchable tree index *I* for data owner. Upon receiving the trapdoor *TD* from the data user, the cloud server executes search over the index tree *I*, and finally returns the corresponding collection of top- *k* ranked encrypted documents. Besides, upon receiving the update information from the data owner, the server needs to update the index *I* and document collection *C* according to the received information. The cloud server in the proposed scheme is considered as “honest-but-curious”, which is employed by lots of works on secure cloud data search

3.4 Rank Search Module

These modules ensure the user to search the files that are searched frequently using rank search. This module allows the user to download the file using his secret key to decrypt the downloaded data. This module allows the Owner to view the uploaded files and downloaded files. The proposed scheme is designed to provide not only multi-keyword query and accurate result ranking, but also dynamic update on document collections. The scheme is designed to prevent the cloud server from learning additional information about the document collection, the index tree, and the query.

IV. Algorithm

4.1 KNN algorithm: To demonstrate a *k*-nearest neighbor analysis, let's consider the task of classifying a new object (query point) among a number of known examples. This is shown in the figure below, which depicts the examples (instances) with the plus and minus signs and the query point with a red circle. Our task is to estimate (classify) the outcome of the query point based on a selected number of its nearest neighbors. In other words, we want to know whether the query point can be classified as a plus or a minus sign [12].

4.2 Privacy-Preserving Primitives: In order, to create a privacy-preserving version of k-means that does not use a TTP we have to devise a privacy-preserving protocol to compute the cluster means. Consider the computation of a single cluster mean µi. Recall that in distributed k-means each party sends (ai, bi) and (di, ei) to the TTP, which computes; this is precisely the function for which we have to devise a privacy-preserving protocol.

4.3 Secure Harsh Algorithm: SHA-2is a set of cryptographic hash functions designed by the National Security Agency (NSA) SHA stands for Secure Hash Algorithm. Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their collision resistance [13].

V. Conclusion

In our project we propose a key secure for the Access between user and the owner. Normally the cloud sever is not secure with the Access permission using key generation. Key generation is to secure the data in cloud server to access by the user. Here we are using K-Nearest Neighbours Algorithm to classify the data to the required user. The access key is generated by Admin to secure the data and for the quick access. In this paper we describe and solve the problem of multikey word ranked search over encrypted cloud data, and set up a range of privacy requirements. Among various multi-keyword semantics, we select the efficient similarity measure of “coordinate matching,” i.e., as many equivalent as possible, to effectively capture the relevance of outsourced documents to the query Keywords, and utilize “inner product similarity” to quantitatively calculate such comparison measure. In order to acquire the test of supporting multi-keyword semantic without privacy violation, we offer a basic idea of MRSE using secure inner product calculation. We plan to investigate alternative and more efficient solutions to this problem in our future work. Also, we will investigate and extend our research to other classification algorithms.

References

[1]. P. Mell and T. Grance, “The NIST definition of cloud computing (draft),”NIST Special Publication, vol. 800, p. 145, 2011.

[2]. S. De Capitani di Vimercati, S. Foresti, and P. Samarati, “Managing and accessing data in the cloud: Privacy risks and approaches,” in Proc. 7th Int. Conf. Risk Security Internet Syst., 2012, pp. 1–9. 1272 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 5, MAY 2015

[3]. P. Williams, R. Sion, and B. Carbunar, “Building castles out of mud: Practical access pattern privacy and correctness on untrusted storage,” in Proc. 15th ACM Conf. Comput. Commun. Security, 2008.

[4]. P. Paillier, “Public key cryptosystems based on composite degree residuosity classes,” in Proc. 17th Int. Conf. Theory Appl. Cryptographic Techn., 1999, pp. 223–238.

[5]. [5] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, “k-nearest neighbor classification over semantically secure encrypted relational data,” eprint arXiv:1403.5001, 2014.

[6]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,” Proc. IEEE INFOCOM, pp.829- 837, Apr, 2011.

[7]. L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M.Lindner, “A Break in the Clouds: Towards a Cloud Definition,” ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50-55, 2009.

[8]. N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, “LT Codes-Based Secure and Reliable Cloud Storage Service,” Proc. IEEE INFOCOM, pp. 693701, 2012. *4+ S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. 14th Int’l Conf. Financial Cryptograpy and Data Security, Jan. 2010. *5+ A. Singhal, “Modern Information Retrieval: A Brief Overview,” IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35- 43, Mar. 2001.

[9]. I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishing May 1999. *7+ D. Song, D. Wagner, and A. Perrig, “Practical Techniques for Searches on Encrypted Data,” Proc. IEEE Symp. Security and Privacy, 2000.

[10]. E.-J. Goh, “Secure Indexes,” Cryptology ePrint Archive, http:// eprint.iacr.org/2003/216. 2003.

[11]. M. Mitzenmacher, “Privacy Preserving Keyword Searches on Remote Encrypted Data,” Proc. Third Int’l Conf. Applied Cryptography and Network Security, 2005.

[12]. R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions,” Proc. 13th ACM Conf. Computer and Comm. Security (CCS ’06), 2006.

[13]. D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key Encryption with Keyword Search,” Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2004.