15th International Conference on Recent Trends in Engineering, Applied Science and Management
Gurukul Institute of Engineering & Technology, Rajasthan, India. ICRTEASM-18
1st April 2018 www.conferenceworld.in ISBN: 978-93-87793-18-7

Introduction to Semantic Search Over Encrypted Data on Cloud

Dipti D. Mehare[1], Prof. A. V. Deorankar[2]

[1]P.G. Scholar, Department of Computer Science and Engineering, Government college of Engineering, Amravati, Maharashtra, (India)

[2]Assistant Professor, Department of Computer science and Engineering, Government college of Engineering, Amravati, Maharashtra, (India)

ABSTRACT:

Cloud storage becomes more and more popular in the recent trend since it provides various benefits over the traditional storage solutions In today's world large amount of data can be stored on cloud. Therefore security over the cloud becomes a major issue. To reduce the problem related to the security of sensitive data, it is preferable to store data in encrypted form. Encrypted data protects the data against illegal access. In this paper, we propose an efficient scheme for semantic search over encrypted data. The vector space model and TFIDF model are used to construct index and query generation. The KNN algorithm used to encrypt index and query vectors. We construct a special tree called Balanced M-way Search (BMS) Tree for indexing and BMS tree takes sub-linear time complexity.

Keywords: The Cloud Computing privacy preserving, semantic keyword search, BMS tree.

I INTRODUCTION

With the advent of cloud computing, outsourcing storage has become an increasingly popular trend and many users are storing their data in the cloud in order to benefit from unlimited storage space. Some of this data is sensitive and must be protected from unauthorized access, including from cloud operators, classically considered honest-but-curious. A classical solution in this setting is for users to encrypt their data before sending them to the storage server. This solution protects the data but is at odds with the utility of the cloud, as the data becomes completely obfuscated and the cloud operator cannot extract any information from the data nor perform searches on it.

Data encryption restricts user's ability to perform keyword search. It also demands the protection of keyword privacy because few keywords contains important information about data files. For the effective search system, efficient and flexible searchable scheme is semantic based search. Vector space model

(VSM) is used to build document index to address multi keyword search and result ranking. Each document is expressed as a vector where each dimension value is the term frequency (TF). The vector as the same dimension with document index where each dimension value is the inverse document frequency (IDF) weight. The K-nearest neighbor's algorithm (KNN) is a non-parametric method used for classification and regression. The output of KNN algorithm depends on whether KNN is used for classification or regression. We construct a special tree called Balanced M-way Search (BMS) Tree for indexing and BMS tree takes sub-linear time complexity.

II SYSTEM ARCHITECTURE

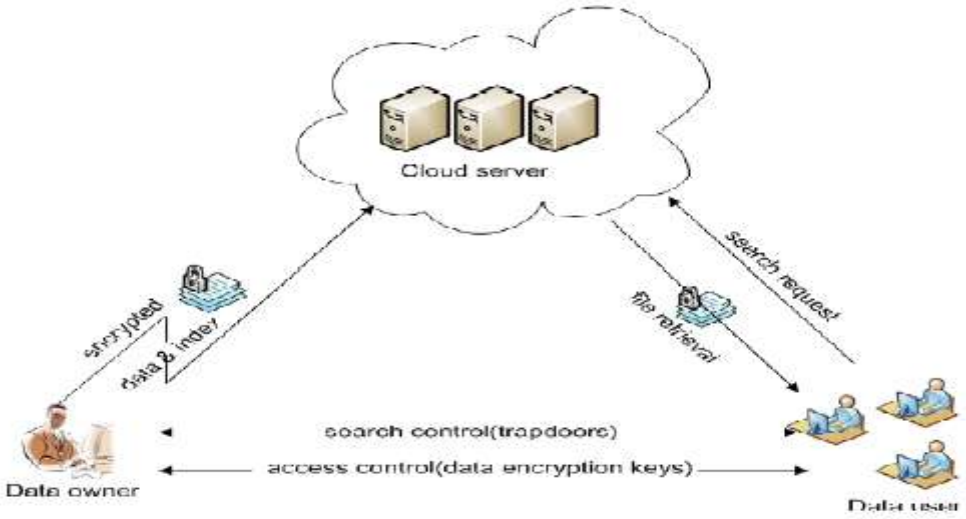

Fig No.1 Architecture of Semantic Search Over Encrypted Data

A cloud computing system model involves three different entities. Those are Data Owner, Cloud Service Provider and Data user as illustrated in Fig. 1. The responsibility of each entity is as follows:

Data Owner (DO): DO has a collection data documents DC= {d1, d2…, dm} with sensitive information to be outsourced to the cloud server. To provide data privacy, the documents are encrypted before outsourcing. DO creates a dictionary based on keywords extracted from the all m documents based on Term Frequency Inverted Document Frequency (TFIDF). The dictionary includes synonyms for each keyword.

Data users: Data users are the users who accessing sensitive data from the cloud. The cloud server searches keywords or synonyms related to documents, which are interested to data user and sends to the data owner.

Cloud Service Provider (CSP): Cloud server receives encrypted documents and encrypted index vectors from data owner and stores into data owner's cloud storage.

III ENCRYPTED DATA SEARCH

The Vector Space Model And Keyword Extraction: The representation of a pool of documents as vectors in a common vector space is called vector space model. This model along with TF-IDF is used in plain text information retrieval. TF-IDF stands for term frequency-inverse document frequency, and the TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

Here, TFt,d is the term frequency defined as the number of times keyword (term) t appears in the document d. The number of documents that contains keyword t is known as document frequency (DFt). Inverse Document Frequency (IDFt) of a keyword t obtained fromthe total number of documents is divided by document frequency. Every document file is denoted with a vector, whose elements are the normalized TF weights of keywords in this document file. The query is also denoted with a vector, whose elements are normalized IDF weights of query keywords in the document collection. Naturally, the size of the query vector and document vector are equal to the total number of keywords in the dictionary. The dot product between query vector and document vector gives the relevance score of the corresponding document; it quantifies the relevance between the query and the corresponding document.

TFt,dweight (wt,d) = log (TFt,d)

IDFtweight (wt,q)= log (N/DFt)

Normalized TFt,d weight = $wt,d\sqrt{\Sigma(wt,d)2Nt}$**=1**

Normalized IDFt weight = $wt,\sqrt{\Sigma(wt,q)2Nt}$**=1**

Relevance score of query vector Q and document vector or index vector Fu of node uis calculated as follows:

Score (Fu, Q) = Fu . Q = $\Sigma TFu,\times IDFtt\in Wq$ **Where TFu,t is the TF value of term t at node u**

BMS Tree Index Construction: In index tree construction process, we generate node for every document in the document dictionary. All these node act as a leaf node. After this internal node are formed based on the leaf nodes. Index tree construction process described in the algorithum given below.

Algorithm 1 BuildBMSIndexTree(DC):

For each data document Ddid in DC do

Construct leaf node l for Ddid l.ID=GenID(), l.child[i]=null for i=1,…, b; l.DID=DID, and F[i]=TFDdid,ki for i=1,…, n;

Insert l to CurrentNodeCollection;
End for
While the number of nodes in CurrentNodeCollection is more than 1 do
For each five of nodes u1, u2, u3, u4, and u5 in CurrentNodeCollection do
Generate a parent node u for u1, u2, u3, u4, and u5with u.ID=GenID(), u.child[i]= uifor i = 1to 5;
u.DID = 0, and D[i] = max{ui.F[j] for i=1 to 5} for each j=1 to n;
Insert u to TempNodeCollection;
End for
The remaining nodes (less than 5 nodes) in CurrentNodeCollection generate a parent node u like above;
Insert u to TempNodeCollection;
Replace CurrentNodeCollection with TempNodeCollection and then free the TempNodeCollection;
End while
Return only one node, left in the CurrentNodeCollection called the root node;

Example:

We constructed index tree on the plaintext, The data structure of the node is defined as (ID, F, child [], DID), where ID is a unique id generated using GenID() function, F is index vector, child[] is pointers to children of the node and DID is a document ID. In the algorithm, we used two variables CurrentNodeCollection and TempNodeCollection to store collection of nodes. CurrentNodeCollection stores the set of currently processing nodes which have no parents and TempNodeCollection stores set of newly formed nodes. Fu[i] always stores the biggest TF value of wiamong its children. The possible largest relevance score of its children is estimated using this technique.BMS Index tree for our scheme shown below,

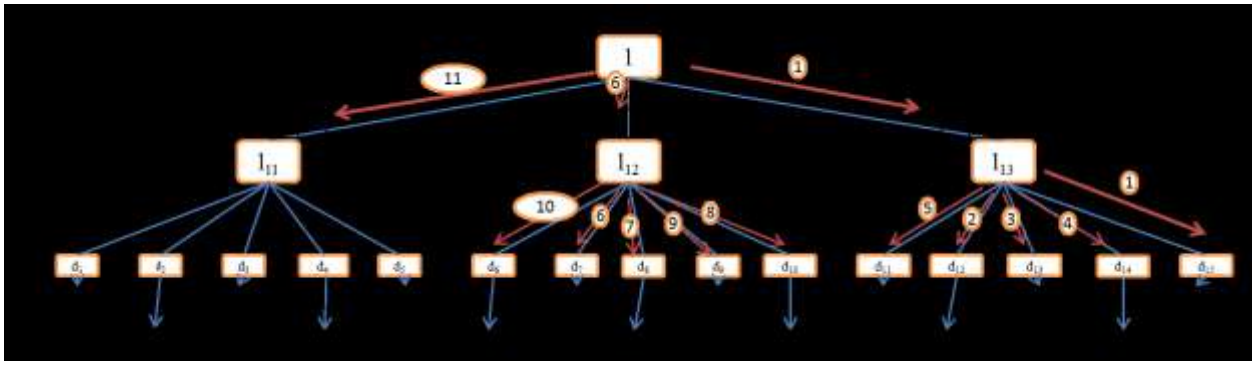

Fig No. 2: An example of BMS tree and search process

IV CONCLUSIONS

In this paper, we proposed efficient semantic search method Over Encrypted Data on Cloud. The data owner who wishes to keep their file hidden from other user will provide secrete key to user for searching file stored on separate folder and will be displayed only to that particular user. When the data owner provides the access policy to his uploaded file then the file will displayed to all data user who ever generates search query related to that file semantics. The data user will be provided the facility of rating the document being viewed. Once user provides the search query, all the documents that hold the exact query keyword or the semantically related keyword are listed. The aim of adding all the features to the cloud search service is that the cloud consumers can search the most relevant products or data by using the designed system.

REFERENCES

[1]. Veerraju Gampala, Sreelatha Malempati**, "** An efficient Multi-Keyword Synonym Ranked Query over Encrypted Cloud Data using BMS Tree",International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 1 (2016)pp 738-743.

[2]. K VENINGSTON, R SHANMUGALAKSHMI and V NIRMALA, "Semantic association ranking schemes for information retrieval applications using term association graph representation" *Sa¯dhana¯* Vol. 40, Part 6, September 2015, pp. 1793–1819._c Indian Academy of Sciences.

[3]. Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu," Efficient Similarity Search over Encrypted Data" *Department of Computer Science, The University of Texas at Dalla Richardson, TX 75080, USA.*

[4]. Tarik Moataz_, Abdullatif Shikfay, Nora Cuppens-Boulahia_, and Fr´ed´eric Cuppens, " Semantic Search Over Encrypted Data" ,TELECOM Bretagne, Rennes, France.

[5]. Avani Konda, Sai Praneeth Gudimetla, Balaji T, Gopi Krishna Subramanyam, Usha Kiruthika " Synonymous Keyword Search Over Encrypted Data in Cloud" International Journal of MC Square Scientific Research Vol.9, No.2 2017.

[6]. Jyoti Gautam, Ela Kumar, and Mehjabin Khatoon," Semantic Web Improved with IDF Feature of the TFIDF Algorithm" Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I, IMECS 2014, March 12 - 14, 2014, Hong Kong.

[7]. Mrs. Jyoti Gautam, and Dr. Ela Kumar, " Semantic Web Improved with the Weighted IDF Feature" International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, 2015.

[8]. Ankur Verma, Bhagyashri Patki, Priyanka Sawant , Prajkta Waingankar † and Dike Onkar D," Efficient Similarity Search over Encrypted Data on Cloud" International Journal of Current Engineering and Technology.

[9]. Ho, Kam Ho, "Semantic Search over Encrypted Data in Cloud Computing" (2013).Master's Projects. 347.