Secure Multi-keyword Ranked Search Over Encrypted Cloud Data

[1]Katre Sharad, [2]Hipparkar Pradip, [3]Sagare Ravi, [4]Hiwale Sumit, [5]Smita Bhosale
[1,2,3,4,5]D. Y. Patil College of Engineering, Ambi, Pune, Maharashtra, India

Abstract

A Secure and Dynamic Multi-keyword Search the increasing popularity of cloud computing, more and more data owners. They are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. The sensitive data encrypted before outsourcing for privacy requirements. Which obsoletes data utilization like keyword-based document retrieval. Here a secure multi-keyword ranked search scheme over encrypted cloud data is presented. Which simultaneously supports dynamic update operations like that deletion and insertion of documents. The vector space model used to TFIDF model are combined, for index construction and query generation. This construct a special tree-based index structure and propose a "GDFS" algorithm is used to efficient multi-keyword search. The secure KNN algorithm is utilized to encrypt the index and query vectors. Also ensure accurate relevance score calculation between encrypted index and query vectors. To resist statistical attacks, phantom terms are added to the index vector for blinding search results. The use of our special tree-based index structure, for achieve sub-linear search time and deal with the deletion and insertion of documents with flexibility. Extensive experiments are conducted to demonstrate the efficiency of search scheme.

General Terms

Searchable Encryption, Multi-Keyword Ranked Search, Dynamic Update, Cloud Computing.

Keywords

CSP-Cloud Service Providers, GDFS-Greedy Depth First Search, KNN-k-Nearest Neighbors algorithm

I. Introduction

The Increasing Popularity of use of cloud computing, data owners are aware to outsource their sensitive and complex data management system from local sites to commercial public cloud savings. For protecting the privacy of data, the sensitive data must have to be encrypted before uploading or saving on the cloud. Most of the current systems are works on plain text keyword search. The use of plain text can decrease the privacy of data, So the encrypted cloud data search is the most important than plain text keyword search over cloud data. But considering the large number of data owners, documents and data users in the cloud, it is necessary to allow multiple keyword search requests and in response returns of documents in order of their importance of keyword search. In this paper for the first time defining and solving the challenging problems of a secure and dynamic multi-key search over encrypted cloud data in cloud computing and at the same time it supports dynamic update operations like deletion and insertion of documents. The proposed scheme can achieve sub-linear search time with the deletion and insertion of documents flexibly.

II. Problem Statement

Design and implement an efficient and flexible Secure and dynamic multi-key ranked search scheme over encrypted cloud data using Greedy DFS Algorithm to provide an efficient multi-key ranked search. Use the secure KNN algorithm to encrypt the index and query vectors.

III. Proposed System

In a Secure and Dynamic Multi-keyword Search over Encrypted Cloud Data constructed a special tree-based index structure and used a "Greedy DFS" algorithm for delivery of efficient multi-keyword ranked search. The proposed system can realize sub-linear search time and deal with the deletion and insertion of documents flexibly. Wide-ranging experiments are shown to demonstrate the efficiency of the proposed scheme.

- Plentiful works have been proposed under different risky models to accomplish various search functionality,
- Newly, some dynamic schemes have been proposed to support inserting and deleting operations on text collection.

This paper proposes a secure tree-based search system over the encrypted cloud data, which supports multi keyword search and dynamic operation on the document collection.

IV. Scope of Project

Given system built a special tree-based index structure and suggest a "GDFS" algorithm to offer effective multi-keyword search. The proposed system can realize sub-linear search time and deal with the deletion and insertion of documents openly. Extensive experiments are conducted to demonstrate the efficiency of the proposed system.

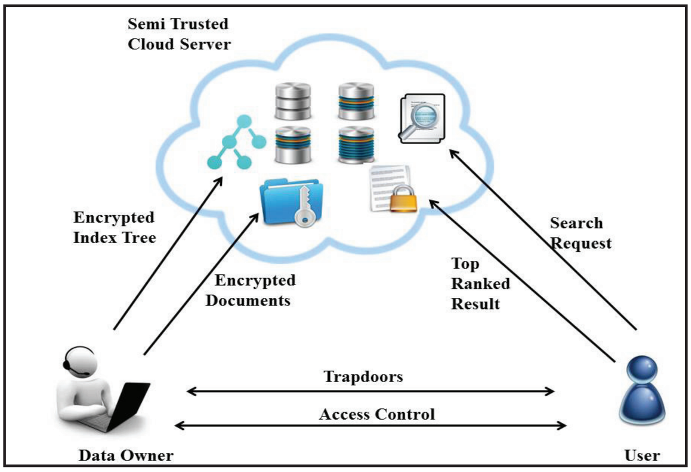

Fig. 1: System Architecture

V. Notations and Preliminaries

- W – The dictionary, namely, the set of keywords, denoted as W = {w1, w2,..., wn}.
- n– The total number of keywords in W.
- Wq – The subset of W, signifying the keywords in the query.
- F – The plaintext document collection, denoted as a collection of n documents F = {f1, f2... fm}. Each document f in the collection can be considered as a sequence of keywords.
- m – The total number of documents in F.
- C – The encrypted document collection kept in the cloud server, represented as C = {c1, c2,..., cm}.

- T – The unencrypted form of index tree for the whole document collection F.
- I – The searchable encrypted tree index generated from T .
- Q – The query vector for keyword set Wq.
- td – The encrypted form of Q, which is called as trapdoor for the search request.
- Du – The index vector deposited in tree node u whose dimension matches to the cardinality of the dictionary W. Note that the node u can be also a leaf node or an internal node of the tree.

I_u – The encrypted form of D_u.

VI. Vector Space Model and Relevance Score Function

Vector space model beside with tf×idf rule is broadly recycled in plaintext information recovery, which efficiently supports multi-keyword search. Here, the term frequency (tf) is the number of times a specified term (keyword) looks within a document, and the inverse document frequency (idf) is realized by dividing the cardinality of document collection by the number of documents covering the keyword. In the vector space model, every document is represented by a vector, whose elements are the normalized 'tf' values of keywords in this document. Every query is also denoted as a vector Q, whose elements are the normalized 'idf' values of query keywords in the document collection. Naturally, the distances of both the 'tf' vector and the 'idf' vector are equivalent to the entire number of keywords and the dot product of the 'tf' vector Du and the 'idf' vector Q can be calculated to quantify the relevance between the query and corresponding document. Below the notations are presented which are used in our relevance evaluation function:

- $N_{f,wi}$ – The number of keyword w_i in document f.
- N – The total no. of documents.
- N_{wi} – The no. of documents that contain keyword w_i.
- $tf'_{f,wi}$ – The tf value of w_i in document f.
- idf'_{wi} – The idf value of w_i in document collection.
- $tf_{u,wi}$ – The normalized 'tf' value of keyword w_i stowed in index vector D_u.
- idf_{wi} – The normalized 'idf' value of keyword w_i in document collection.

The relevance evaluation function is defined as:
RScore

$$(D_u, Q) = D_u \cdot Q = \sum_{w_i \in \mathcal{W}_q} TF_{u,w_i} \times IDF_{w_i} \qquad (1)$$

If u is an internal node of the tree, $tf_{u,wi}$ is calculated from index vectors in the child nodes of u. If the u is a leaf node, $tf_{u,wi}$ is calculated as:

tfu,wi = tf′f,wi √∑wi∈W(tf′f,wi)2 (2)

where tf′f,wi = 1+lnNf,wi. And in the search vector Q, idfwi is calculated as:

idfwi = idf′wi √∑wi∈Wq(idf′wi)2 (3)

where idf′wi = ln(1 + N/Nwi).

Keyword Balanced Binary Tree. The balanced binary tree is broadly used to deal with optimization problems. The keyword balanced binary (KBB) tree in our system is a dynamic data structure whose node stores a vector D. The elements of vector D are the normalized TF values. Sometimes, we refer the vector D in the node u to Du for simplicity. Formally, the node u in our KBB tree is defined as follows:

u = ⟨ID,D,Pl,Pr,FID⟩ , (4)

Where, ID denotes the identity of node u, Pl and Pr are respectively the pointers to the left and right child of node u. If the node u is a leaf node of the tree, FID supplies the identity of a document, and D indicates a vector consisting of the normalized tf values of the keywords to the document. If the node u is an internal node, FID is set to null, and D denotes a vector containing the tf values which is calculated as follows:

D[i]= max{u.Pl → D[i],u.Pr → D[i]},i = 1,...,m. (5)

VII. Acknowledgments

Our thanks to the experts who have contributed towards development of the template.

VIII. Conclusion

We have to remake the list and convey the new secure keys to all the approved users. Secondly, symmetric SE conspires more often than not expect that all the information clients are dependable. It is not reasonable and an exploitative information client will prompt numerous protected issues. For instance, an untrustworthy information client may look the archives and disseminate the decoded records to the unapproved ones.

The parallel search process can be carried out to reduce the time cost. The security of the scheme is used to protect against two threat models by using the secure KNN algorithm.

References

[1] K. Ren, C.Wang, Q.Wang et al.,"Security challenges for the public cloud," IEEE Internet Computing, Vol. 16, No. 1, pp. 69–73, 2012.

[2] S. Kamara, K. Lauter,"Cryptographic cloud storage," In Financial Cryptography and Data Security. Springer, 2010, pp. 136– 149.

[3] C. Gentry,"A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search," In Advances in Cryptology- Eurocrypt 2004. Springer, 2004, pp. 506–522.

[5] D. Boneh, E. Kushilevitz, R. Ostrovsky, W. E. Skeith III,"Public key encryption that allows pir queries," In Advances in Cryptology-CRYPTO 2007. Springer, 2007, pp. 50–67.

[6] D. X. Song, D. Wagner, A. Perrig,"Practical techniques for searches on encrypted data," In Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44–55.

[7] Y.-C. Chang, M. Mitzenmacher,"Privacy preserving keyword searches on remote encrypted data," In Proceedings of the Third international conference on Applied Cryptography and Network Security. Springer-Verlag, 2005, pp. 442–455.

[8] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," In Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp.79–88.

Katre Sharad, D. Y. Patil College of Engineering, Ambi, Pune, Maharashtra, India.

Hipparkar Pradip, D. Y. Patil College of Engineering, Ambi, Pune, Maharashtra, India.

Sagare Ravi, D. Y. Patil College of Engineering, Ambi, Pune, Maharashtra, India.

Hiwale Sumit, D. Y. Patil College of Engineering, Ambi, Pune, Maharashtra, India.

Prof. Smita Bhosale, D. Y. Patil College of Engineering, Ambi, Pune, Maharashtra, India.