International Journal of Research In Science & Engineering
Volume: 2 Issue: 3
e-ISSN: 2394-8299
p-ISSN: 2394-8280

MULTI-KEYWORD SEARCHING AND DYNAMIC OPERATIONS ON ENCRYPTED CLOUD DATA

Anita Chavan[1], Supriya Gade[2], Avinash Nagarsoge[3] , Namrata Nikhal [4]

Student, Computer Department, SCOE, anichavan18@gmail.com
[2] Student, Computer Department, SCOE, supriyagade17795@gmail.com
[3] Student, Computer Department, SCOE, avinash.pmcc@gmail.com
[4] Student, Computer Department, SCOE,nikhalnamrata@gmail.com

ABSTRACT

A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data Due to the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. In this paper, we present a secure multi-keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents. Specifically, the vector space model and the widely-used TFIDF model are combined in the index construction and query generation. We construct a special tree-based index structure and propose a "Greedy Depth-first Search" algorithm to provide efficient multi-keyword ranked search. 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. In order to resist statistical attacks, phantom terms are added to the index vector for blinding search results. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.

Keywords*:* Searchable encryption, multi-keyword ranked search, dynamic update, cloud computing.

---

1. INTRODUCTION

1.1 Overview

Cloud computing has been considered as another model of big business IT base, which can sort out gigantic asset of processing, stockpiling and applications, and empower clients to appreciate pervasive, advantageous and on-interest system access to a common pool of configurable registering assets with extraordinary efficiency and negligible monetary overhead. Pulled in by these engaging components, both people and undertakings are spurred to outsource their information to the cloud, rather than acquiring programming and equipment to deal with the information themselves. In spite of the different points of interest of cloud administrations, outsourcing touchy data, (for example, messages, individual wellbeing records, organization finance information, government archives, and so on.) to remote servers brings protection concerns. The cloud administration suppliers (CSPs) that keep the information for clients may get to clients' touchy data without approval. A general way to deal with secure the

information confidentiality is to encode the information before outsourcing. Be that as it may, this will bring about an immense expense regarding information ease of use. For instance, the current systems on magic word based data.

1.2 Architecture:

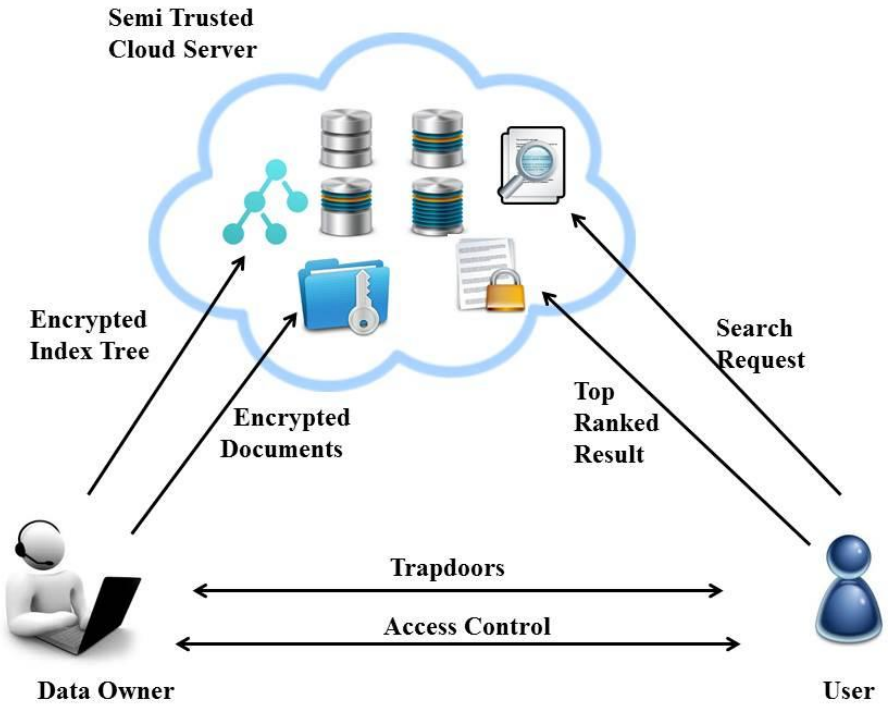

Fig.: Architecture

2. Method of Project Progress and Algorithm ,Index Construction of UDMRS Scheme :

We have briefly introduced the KBB index tree structure, which assists us in introducing the index construction. In the process of index construction, we first generate a tree node for each document in the collection. These nodes are the leaf nodes of the index tree. Then, the internal tree nodes are generated based on these leaf nodes. The formal construction process of the index is presented in Algorithm 1. Note that the index tree T built here is a plaintext.

Following are some notations for Algorithm 1.

Besides, the data structure of the tree node is defined as ⟨ID,D,Pl,Pr,FID⟩, where the unique identity ID for each tree node is generated through the function GenID().

• CurrentNodeSet – The set of current processing nodes which have no parents. If the number of nodes is even, the cardinality of the set is denoted as 2h(h ∈ Z+), else the cardinality is denoted as (2h + 1).

• TempNodeSet – The set of the newly generated nodes. In the index, if Du[i]≠ 0 for an internal node u, there is at least one path from the node u to some leaf, which indicates a document containing the keyword wi.

In addition, Du[i] always stores the biggest normalized TF value of wi among its child nodes. Thus, the possible largest relevance score of its children can be easily estimated.

3. CONCLUSION

A protected, efficient and element inquiry plan is proposed, which bolsters the exact multi-essential word positioned hunt as well as the dynamic erasure and insertion of archives. We build an extraordinary decisive word adjusted

parallel tree as the record, and propose an "Greedy Depth-first Search" calculation to acquire preferred efficiency over straight pursuit. Also, the parallel inquiry procedure can be completed to further diminish the time cost. The plan's security is ensured against two danger models by utilizing the protected kNN calculation. Trial results show the efficiency of our proposed plan. There are still numerous test issues in symmetric SE plans. In the proposed plan, the information proprietor is in charge of producing overhauling data and sending them to the cloud server. Therefore, the information proprietor needs to store the decoded record tree and the data that are important to recalculate the IDF values. Such a dynamic information proprietor may not be exceptionally suitable for the distributed computing model.

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 and 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, and 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, and 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, and 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.