Secure and Efficient Utilization of Encrypted Cloud Data using Multi-Keyword Ranked Search

Ms. Priyanka
Student (M-Tech), Department of Computer Science & Engineering
Sai Vidya Institute of Technology, Bangalore,
Bangalore, 560012, India
priyanka.sm1991@gmail.com

Mrs. Veena N
Assistant Professor, Department of Information Science & Engineering
Sai Vidya Institute of Technology, Bangalore,
Bangalore, 560012, India
Veena_guruprasad@rediffmail.com

***Abstract*—** Cloud Computing is a technology that provides services to users such as software as a service, platform as a service and storage as a service. These services are provided based on Pay-per-Use basis so these services are cost effective and flexible. Due to this advantage of cloud computing, the individuals as well as the enterprises are getting motivated to shift their local sensitive and huge data management system to cloud storage. But the sensitive data has to be encrypted before outsourcing in order to provide security to the data. After the data has outsourced it has to be utilized efficiently without losing the originality as it was stored. In this paper we provide a mechanism called "Multi-keyword Ranked Search over Encrypted cloud data" that gives better and efficient searched result over the encrypted data taking multiple keywords as query, which obsoletes the tradition searching scheme based on plain text search. And we use a "Coordinate Matching" technique to find as many matches as possible and use "inner product similarity" to retrieve relevance search results. So if user wants to retrieve the data stored on cloud, he can specify the multiple keywords and rank for relevance retrieval of results. Finally results the user with top ranked files.

***Keywords*—** *Privacy-preserving Search; Multi-keyword Search; Coordinate matching; Encrypted Cloud computing.*

*****

I. INTRODUCTION

Cloud computing is a type of computing system in which various hardware, software and applications are made to share their facilities over the internet. Cloud computing is the long imagined vision of computing as a utility, where all the cloud customers can store their data into the cloud remotely. And they can enjoy the high quality applications and services from a shared pool of configurable computing resources according to their demand [1], [2]. So cloud provides great flexibility in services and economic savings. This is one of the most important feature of cloud that attracting and motivating both individuals and many enterprises to outsource their local complex data that has become difficult to manage by the management system into the cloud, especially when the data produced by them that has to be stored and utilized easily. To protect the data privacy and unauthorized accesses in cloud and beyond, sensitive data, like, emails, personal health records, financial transactions, tax documents, etc., must be encrypted by data owners before outsourcing their to the public cloud [2]; this, however, obsoletes the traditional data utilization service which was based on plaintext keyword search. Apart from eliminating the local storage management, storing the complex data into the cloud serves is no longer useful if the stored data cannot be easily searched and utilized when needed. Thus, exploring a secure and effective search service over encrypted data stored on cloud is of paramount importance. Considering a situation in which the potentially large number of data users who require services on demand and huge amount of data documents outsourced in cloud, this problem is challenging as it is extremely difficult to meet also the requirements of system usability, and its performance, and scalability.

On the one hand, to meet the requirement of effective data retrieval, the large amount of documents stored on the cloud the cloud server has to perform result relevance ranking, instead of delivering the user with undifferentiated results. Such a ranked search system can enable data users to find the most relevant information quickly, instead of burdensomely sorting through each and every match in the content collection [3]. Ranked search can also easily eliminate unnecessary network traffic by sending back user with only the most relevant data, which is worth having in the "pay-per-use" cloud scenario. For privacy protection, like ranking operation, it should not leak any keyword related information and on the other hand, search result accuracy must be improved as well as to make better searching experience by user, it is also important to support multiple keywords search by ranking system, as single keyword search often results undifferentiated, far too coarse result. "Coordinate matching" [4], (as many matches as possible) is an efficient principle for such multi-keyword semantics to fine-tune the relevance result has become challenging.

II. RELATED WORK

According to paper [5] the earlier **Single Keyword Searchable Encryption** schemes usually build an encrypted searchable index so that content of the file is hidden from the server until it gives proper trapdoors that are generated via secret key [2]. It is initially studied by Song et al. [5] in the setting symmetric key and further improvements and the advanced security definitions were given in Goh [6]. The early work on ranked search with security solved. Where keyword search utilizes keyword frequency for ranking of the results, instead of giving undifferentiated results to the user. Since, it supports only single keyword search. In the setting of public key, it has present the first searchable encryption construction, where as anyone with the public key can store the data on the cloud and write on it. But only those with secret key can search the data on the cloud. Solutions with public key are very computationally expensive usually.

According to paper [7], to enrich search functionalities over the encrypted data they proposed **Boolean Keyword Searchable Encryption**. Such schemes give large overhead that are caused by their fundamental primitives, like computation cost for bilinear map, e.g. [8], or communication cost for secret sharing. More general search approaches like, predicate encryption schemes are recently proposed, they support both conjunctive as well as disjunctive search. Conjunctive keyword searches usually returns "all-or-nothing", which means it that search will return only the documents that contains all the keywords contained that are specified in the query; disjunctive keyword search returns all undifferentiated results, which means it that search will return all the documents if it contains any one of the query keyword. In short, none of Boolean keyword searchable encryption schemes which are existing will support multi-keyword search which gives ranked results over the clouds encrypted data by preserving search privacy.

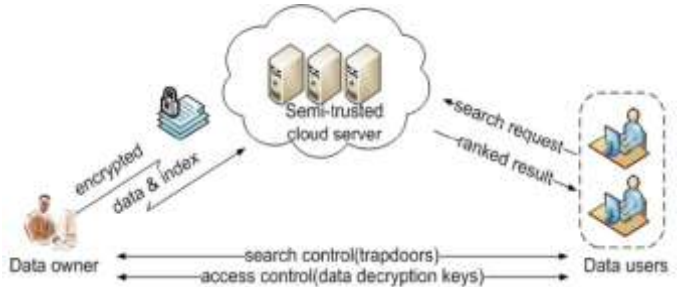

Fig. 1: Architecture proposed system.

III. PROBLEM FORMULATION

A. *System Model*

Consider cloud system model hosting service involving three different entities, as shown in Fig. 1: we have data owner, users, and finally cloud server. Data owner is having number of documents of data as F that he wants to outsource to cloud server. Before outsourcing he will encrypt them in the form C. The outsourced data has to be accessed when needed, so to enable the searching capability over C, data owner first builds an (ESI) encrypted searchable index I from each and every F, and then data owner outsource both the encrypted searchable index I and the encrypted collection of documents C to cloud server. For searching the documents for *t* given keywords, user gets a corresponding trapdoors *T* through some search control mechanisms, like broadcast encryption [8]. After receiving *T* from data users, cloud server take responsibility for searching the index I and then return the user with corresponding set of encrypted documents. In order to improve accuracy of document retrieval, the cloud server should allow data user to specify rank to search result using some ranking techniques (for e.g., coordinate matching). The rank is specified in order to lower the communication costs; user may send a number *k* along with the generated trapdoor *T* so that cloud server only returns back only top-*k* documents which are most relevant to the search query.

B. *Threat Model*

Here cloud server is thought as "honest-but-curious" in this model, which is followed with the most works on the searchable encryptions. Cloud server acts in an "honest" that its fashion and it will correctly follow the designated specification of protocol. And it is "curious" in inferring and analyzes data that is having index in its storage and the message flows which are received during the protocol to learn additional information. Based on type of knowledge cloud server knows, we divide in to two levels of threat models as below as known

- Cipher text Model: In this model, cloud server must only to know dataset C of encrypted form and searchable index I, which are outsourced by data owner. And
- Background Model: In this model, cloud server is expected to possess some background analysis to get more knowledge on the datasets than that of Known cipher text model, like relationship among the given search requests and its related statistical information etc., to identify keywords in the query.

C. *Design Goals*

To achieve the ranked search over the encrypted data so as to experience effective utilization of outsourced cloud data under the defined model, our system design should simultaneously achieve both security and performance guarantees as follows.

- Multi-keyword Ranked Search: for designing search schemes which allow user to enter multi-keyword query instead of giving back the undifferentiated data.
- Privacy-Preserving: To prevent cloud server from learning additional information from dataset and index.
- Efficiency: The designed system gives low communication and computation overhead.

D. Notations

- F – Collection of plaintext document, denoted as a set of m data documents F = (F_1,F_2,...,F_m).
- C – Encrypted document collection which is stored in cloud server, and denoted as C = (C_1,C_2,...,C_m).
- I – searchable index associated with C, denoted as (I_1, I_2, . . ., I_m) where each subindex I_i is built for F_i.
- Wf – the subset of W, represents the keywords in a search request, and denoted as Wf = (W_{j1}, $Wj2$,...,W_{jt}).
- T_W – the trapdoor for the search request Wf.
- F_W –the ranked id list of all documents according to their similarity with Wf.

IV. MRSE Framework

Here we define a framework for MRSE (Multi-Keyword Ranked Search over Encrypted Cloud data) for efficient utilization of out sourced data.

MRSE consists of four steps as follows.

- *Setup(1^l)* Taking a security parameter l as input, data owner outputs a symmetric key as SK.
- *BuildIndex(F, SK)* data owner builds a searchable index I depending on the dataset F, which is encrypted by the symmetric key SK and then outsourced to cloud server. After index construction, the collection of document can be independently encrypted and outsourced.
- *Trapdoor(Wf)* With given t keywords of interest in Wf as input, this algorithm generates a corresponding trapdoor T_We.
- *Query(T_{We}, k, I)* When cloud server receives a query T request as (T_W, k), it performs ranked search over the index I with the help of trapdoor T_W, and finally returns back user F_W, top-k documents of ranked id list sorted by their similarity with *Wf*.

Here search control and access control are not within the scope of this paper. While the former is to regulate how authorized users acquire trapdoors, the later is to manage users' access to outsourced documents.

To achieve efficient multi-keyword ranked search, we employ "inner product similarity" [4] to quantitatively formalize the efficient ranking principle "coordinate matching". consider, for document F_i, we have D_i as a binary data vector where each bit $D_i[j] \in \{0,1\}$ representing the presence of the every related keyword W_j in that data document, and Q is anther binary vector for query indicating the keywords of user interest where each bit $Q[j] \in \{0,1\}$ representing the existence of the corresponding keyword W_j in the query *Wf*. The similarity score between document F_i and query *Wf* is therefore expressed $D_i \cdot Q$ i.e., is inner product of binary column vectors. The cloud server must be given with the capability to compare similarities of different documents of the query. But, the data vector D_i, the query vector Q al well as their inner product $D_i \cdot Q$ should not be exposed to cloud server to preserve strict system-wise privacy.

V. COORDINATE MATCHING

The coordinate matching is a ranking principle that gives the presence of the keyword in the data document or the query is shown as eq.1 in the data vector or the query vector. We have some factors which can make impact on search usability. Such as, when a particular keyword appears in most of the documents in the collection of data sets, then the importance of that keyword in query is significantly less than that of all the other keywords that appears in fewer documents. Similarly, if a particular keyword is present at multiple location of the document then user may prefer this document than that of preferring the other document that contains only single query keyword at only one location. Hence for capturing these kind of information in the searching process, we use (TF _ IDF) some weighting rules within vector space model in order to calculate similarity, weighing rules like TF (term frequency) that is the total number of times the given keyword or term appears within a document and IDF (inverse document frequency) which is considered by dividing the all the number of files that are contained in whole collection of documents with number of files containing the keyword or term. The score for similarity is computed using following equation

$$Score(F_i, Q) = \frac{1}{|F_i|} \sum_{W_j \in \tilde{W}} (1 + \ln f_{i,j}) \cdot \ln\left(1 + \frac{m}{f_j}\right). \quad (2)$$

Here *fi,j:* indicates the term frequency(TF) for the *Wj* keyword in file *Fi*; and *fj:* indicates number of files contain *Wj* keyword and that is called document frequency; and m:

 ISSN: 2321-8169
2474 - 2478

indicates the total number of files contained in the collection; and |*Fi*| indicated euclidean length for file *Fi*, obtained by

$$\sqrt{\sum_{j=1}^{n}(1+\ln f_{i,j})^2},$$

VI. RESUTLS

Results are the output analysis of the proposed system which is shown as below. The Fig.2 shows data owner and user login page, where user and the data owner is register with the cloud server. Both user and data owner has to login by providing credentials.

Fig. 2: Data owner and user login page

Fig.3 shows the data owner uploading or outsourcing the documents to the cloud. First he chooses the file to b uploaded and extracts the all possible keywords and generates the indexes for keywords and encrypts the keywords along with entire document and uploads to cloud server.

Fig. 3: Data owner's file upload page.

Fig. 4 shows the user page where he can give keywords to be searched and also can give the top-k files to be retrieved is shown Fig. 5.

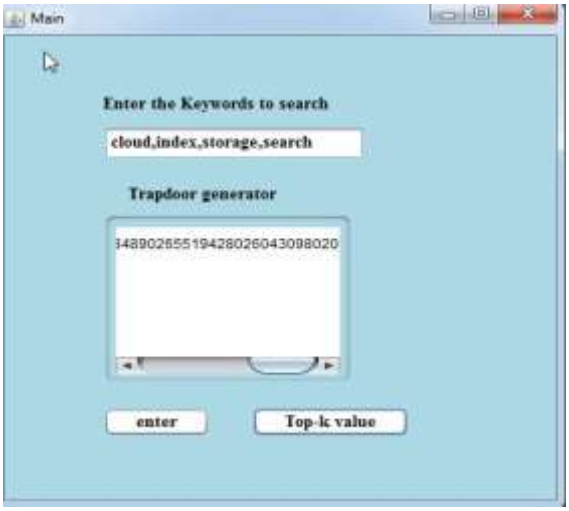

Fig. 4: User's search page.

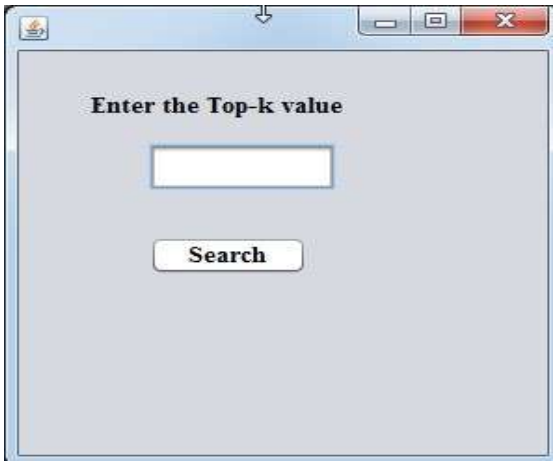

Fig. 5: Entering top-k value page.

Fig. 6 shows the scores generated for matching files and upon receiving top-k value top k files are given to the user.

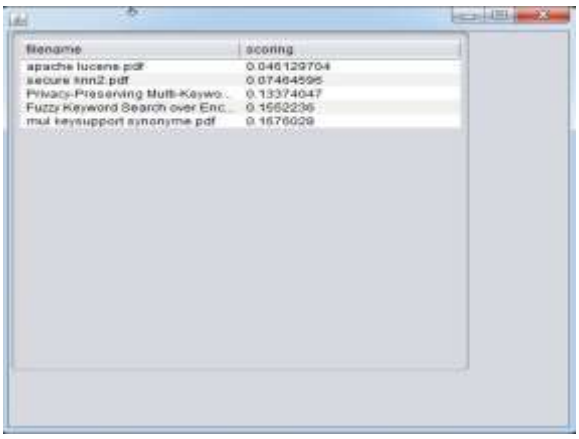

Fig. 6: Scores generated for matching files page.

VII. CONCLUSION

In this paper we propose a scheme called multi-keyword ranked search over encrypted cloud data which gives solution to the problem i.e., secure and efficient utilization of encrypted cloud data using multi-keyword ranked search. And using coordinate matching we find many similarities of query keywords and using inner product similarity of result found we will get the relevance ranked resultant files. There by user experiences better search results and gets the results appropriately by specifying the rank for the results.

REFERENCES

[1] 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.

[2] "Cryptographic cloud storage," by L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, in *RLCPS, January 2010,*

[3] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Engineering Bulletin*, 2001.

[4] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. of ACM CCS*, 2006.

[5] W. Harrower, "Searching Encrypted Data," technical report, Dept. of Computing, Imperial College London, 2009..

[6] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. of ACNS*, 2004, pp. 31–45.

[7] L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches over Encrypted Data," Proc. Seventh Int'l Conf. Information and Comm. Security (ICICS '05), 2005.

[8] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. 12th Int'l Conf. Extending Database Technology, 2009.