# Privacy Preserving Keyword Search over Cloud Data

Pranjali S. Dubey[1], Puja A. Dhakulkar[2], Ankita A. Gaikwad[3], Sneha P. Dhokane[4]

[1,2,3,4] Student, Department of CSE, Des'scoet, Dhamangaon Rly, Maharashtra, India

**Abstract** - *With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. We present a scheme that discusses secure rank based keyword search over an encrypted cloud data. The data that has to be outsourced is encrypted using symmetric encryption algorithm for data confidentiality. The index file of the keyword set that has to be searched is outsourced to the local trusted server where the keyword set that is generated from the data files is also stored. This is done so that any un-trusted server cannot learn about the data with the help of the index formed. The files are listed based on the certain relevance criteria. User requests for the required files to the un-trusted server. The parameters required for ranking is got from the data stored while indexing. Based on the ranking, the files are retrieved from the un-trusted server and displayed to the user. The proposed system can be extended to support Boolean search and Fuzzy keyword search techniques.*

***Key Words***: Symmetric Encryption algorithm, Rank based search, multiple string matching, relevance scoring, privacy preserving, and cloud computing.

## 1. INTRODUCTION

Cloud computing is the long dreamed vision of computing as a utility, where cloud customers can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources [1]. Its great flexibility and economic savings are motivating both individuals and enterprises to outsource their local complex data management system into the cloud. To protect data privacy and combat unsolicited accesses in the cloud and beyond, sensitive data, e.g., emails, personal health records, photo albums, tax documents, financial transactions, etc., may have to be encrypted by data owners before outsourcing to the commercial public cloud [2]; this, however, obsoletes the traditional data utilization service based on plaintext keyword search.

The trivial solution of downloading all the data and decrypting locally is clearly impractical, due to the huge amount of bandwidth cost in cloud scale systems. Moreover, aside from eliminating the local storage management, storing data into the cloud serves no purpose unless they can be easily searched and utilized. Thus, exploring privacy-preserving and effective search service over encrypted cloud data is of paramount importance. Considering the potentially large number of on-demand data users and huge amount of outsourced data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet also the requirements of performance, system usability and scalability. On the one hand, to meet the effective data retrieval need, the large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results. Such ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection [3]. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the "pay-as-youuse" cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information.

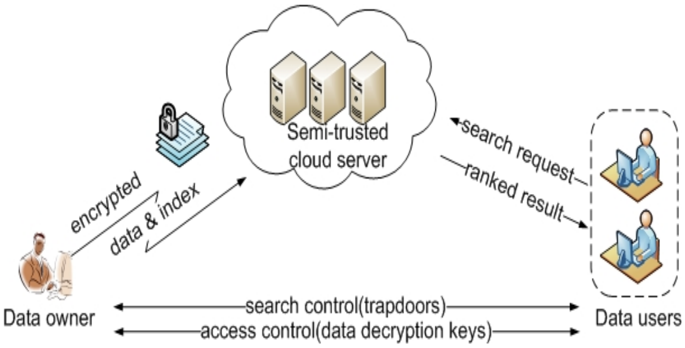

**Fig.1** Diagram Shows Search Over Encrypted Cloud Data

On the other hand, to improve the search result accuracy as well as to enhance the user searching experience, it is also necessary for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse results. As a common practice indicated by today's web search engines (e.g., Google search), data users may tend to provide a set of keywords instead of only one as the indicator of their search interest to retrieve the most relevant data. And each keyword in the search request is able to help narrow down the search result further. "Coordinate matching" [4], i.e., as many matches as possible, is an efficient similarity measure among such multi-keyword semantics to refine the result relevance, and has been widely used in the plaintext information retrieval (IR) community. However, how to apply it in the encrypted cloud data search system remains a very challenging task because of inherent security and privacy obstacles, including various strict requirements like the data privacy, the index privacy, the keyword privacy, and many others.

It brings the software and data to the centralized data centers from where a large community of users can access information on pay per use basis. This poses security threats over the data stored. Data confidentiality may be compromised which has to be taken care of. So it becomes necessary to encrypt the data before outsourcing it to the cloud server. This makes data utilization a challenging task. Traditional searching mechanisms provide Boolean search to search over encrypted data, which is not applicable when the number of users and the number of data files stored in the cloud is large. They also impose two major issues, one being the post-processing that has to be done by the users to find the relevant document in need and the other is the network traffic that is undesirable in present scenario when all the files matching with keywords is retrieved. But this paper proposes ranked keyword search that overcomes these issues.

## 2. RELATED WORK

Searchable encryption is a helpful technique that treats encrypted data as documents and allows a user to securely search through a single keyword and retrieve documents of interest. However, direct application of these approaches to the secure large scale cloud data utilization system would not be necessarily suitable, as they are developed as crypto primitives and cannot accommodate such high service-level requirements like system usability, user searching experience, and easy information discovery. Although some recent designs have been proposed to support Boolean keyword search as an attempt to enrich the search flexibility, they are still not adequate to provide users with acceptable result ranking functionality

### 2.1 Single Keyword Searchable Encryption

Traditional single keyword searchable encryption schemes usually build an encrypted searchable index such that its content is hidden to the server unless it is given appropriate trapdoors generated via secret key(s) [2]. It is first studied by Song et al. [5] in the symmetric key setting, and improvements and advanced security definitions are given in Goh [6], Chang et al. [7] and Curtmola et al. [8]. Our early work [22] solves secure ranked keyword search which utilizes keyword frequency to rank results instead of returning undifferentiated results. However, it only supports single keyword search. In the public key setting, Boneh et al. [9] present the first searchable encryption construction, where anyone with public key can write to the data stored on server but only authorized users with private key can search. Public key solutions are usually very computationally expensive however. Furthermore, the keyword privacy could not be protected in the public key setting since server could encrypt any keyword with public key and then use the received trapdoor to evaluate this ciphertext.

### 2.2 Boolean Keyword Searchable Encryption

To enrich search functionalities, conjunctive keyword search [14]–[18] over encrypted data have been proposed. These schemes incur large overhead caused by their fundamental primitives, such as computation cost by bilinear map, e.g. [16], or communication cost by secret sharing, e.g. [15]. As a more general search approach, predicate encryption schemes [19]–[21] are recently proposed to support both conjunctive and disjunctive search. Conjunctive keyword search returns "all-or-nothing", which means it only returns those documents in which all the keywords specified by the search query appear; disjunctive keyword search returns undifferentiated results, which means it returns every document that contains a subset of the specific keywords, even only one keyword of interest. In short, none of existing Boolean keyword searchable encryption schemes support multiple keywords ranked search over encrypted cloud data while preserving privacy as we propose to explore in this paper. Note that, inner product queries in predicate encryption only predicates whether two vectors are orthogonal or not, i.e., the inner product value is concealed except when it equals zero. Without providing the capability to compare concealed inner products, predicate encryption is not qualified for performing ranked search. Furthermore, most of these schemes are built upon the expensive evaluation of pairing operations on elliptic curves. Such inefficiency disadvantage also limits their practical performance when deployed in the cloud. On a different front, the research on top-k retrieval [27] in database community is also loosely connected to our problem.

### 2.3 Fuzzy Keyword Search

The traditional searching techniques retrieve files based on exact keyword match only but Fuzzy keyword search technique extends this feature by supporting common typos and format inconsistencies that occurs when the user types the keywords. The data privacy that is maintained during exact keyword search is ensured when this method is used. Wild card based technique [4] is used to create efficient fuzzy keyword sets that are used for matching relevant documents. The keyword sets are created using Edit Distance algorithm that quantifies word similarity. These keyword sets reduce storage and representation overhead by eliminating the need to generate all fuzzy keywords, rather generating on similarity basis. The search result that is provided is based on a fuzzy keyword data set that is generated whenever the exact match search fails.

## 3. PROPOSED SYSTEM

We have proposed an efficient scheme which enables the Cloud Service Provider (CSP) to determine the files that are related to the keywords searched by the user rank them and send the most relevant files without knowing any information about the cloud. Our schema consists of three entities: Data owner, Un-trusted cloud server and local trusted server. The data owner is the one whose data is stored in cloud server and he is also authorized to search over his files. Cloud server is an un-trusted server which provides storage service where data owners store their documents in encrypted form. The trusted local server stores the index that is created for the files. The system architecture is shown in Fig 1. We assume that authorization of users and keys used for encryption are managed by the local trusted server.

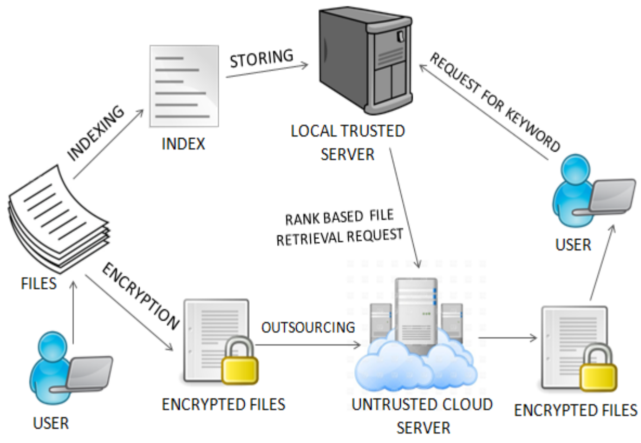

**Fig.2** Proposed System Architecture

### ➢ Encryption Algorithm

As we are not going to perform any operation on the outsourced files to search of the keywords, we can use any of the existing light weight symmetric key Encryption algorithms and unload the data files to the cloud. We use DES to encrypt the file and then outsource it.

### ➢ String Matching Algorithm

Aho-Corasick is found to be the efficient algorithm for multiple string matching that finds all occurrences of the pattern present in the files that are to be outsourced to the un-trusted cloud server. The algorithm consists of two parts:

The first part is building of the tree (trie) from keywords we want to search for, and the second part is searching the test for the keywords using the previously built tree. The tree is a finite state machine, which is a deterministic model of behavior composed of finite number of states and transitions between those states. In the first phase of tree building, keywords are added to the tree where the root node is just a place holder and contains links to other letters. A trie is the keyword tree for a set of keywords K is a rooted tree T such that each edge of T is labeled by a character and any two edges out of a node have different labels.

### ➢ Indexing

Index is created as a list of mappings [10] which correspond to each keyword. The list for a particular keyword contains details such as: 1. File ids of the files which has the particular keyword 2. Term frequency for each file which denotes the number of times the keyword has occurred in the file. This measures the importance of the keyword in that file. 3. Length of each file 4. Relevance score for each file 5. Number of files that has the particular keyword Data structures such as B+ trees can be used to store this data. Term frequency, length of the file, number of files for the keyword are used to calculate the relevance score for each file by scoring mechanisms which is discussed later in the Ranking modules.

The previous papers discuss architectures [5][6] where both the index and the files are stored in encrypted form in the un-trusted server. Whenever user searches for a word, the request is sent to the un-trusted server, which searches over the index and sends the entire mapping that is created for the word to the user. The user has the overhead to decrypt and request to retrieve the most relevant files based on the relevance score information in the index. This takes up a huge amount of bandwidth and round trip time. To reduce the overheads, a new architecture that stores the index as plain text in the local trusted server is proposed. When user searches for a word, the word is sent to the local trusted server, which searches the index, finds out the most relevant files and requests un-trusted server for the files to be retrieved and sent to user thereby ensuring data confidentiality in un-trusted server.

### ➢ Ranking

Once the documents are stored and indexed, the next important function is to rank them using details available such that the user retrieves the top „k‟ most relevant documents. To do so, we need to calculate a numeric score for each file. In the IR community, the most widely used ranking functions are based on the TF X IDF rule, where TF stands for Term frequency which represents the number of times a keyword is present in a file and IDF stands for Inverse Document Frequency which is defined as the ratio of number of file containing the word to the total number of files present in the server.

## 4. CONCLUSION

In this paper, we define and solve the problem of multi-keyword ranked search over encrypted cloud data, and establish a variety of privacy requirements. Among various multi-keyword semantics, we choose the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to effectively capture the relevance of outsourced documents to the query keywords, and use "inner product similarity" to quantitatively evaluate such similarity measure. The scheme generates indexes that help the user to search for his documents in a secure environment. The files matching the keyword search are further ranked based on the relevant score calculated with term frequency, file length etc.

## 5. REFERENCES

[1] 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.

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in RLCPS, January 2010, LNCS. Springer, Heidelberg.

[3] A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001

[4] I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.

[5] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of S&P, 2000.

[6] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, 2003, http:// eprint.iacr.org/2003/216.

[7] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS, 2005.

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

[9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYPT, 2004.

[10] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in Proc. of CRYPTO, 2007.

[11] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions," J. Cryptol., vol. 21, no. 3, pp. 350–391, 2008.

[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, March 2010.