An Efficient and Secure Multi-keyword Ranked Search over Encrypted Cloud

Mr. Balasaheb B. Jadhav[1], Prof. Vinod S. Wadne[2]
[1, 2] Department of Computer Engineering
[1, 2] ICOER, Wagholi, Pune University

Abstract- *Due to the increasing storage and computing requirements of users, everyday huge amount of data is outsourced to remote, but not necessarily trusted servers. There are several privacy issues regarding to accessing data on such servers; two of them can easily be identified: sensitivity of i) keywords sent in queries and ii) the data retrieved; both need to be hidden. We propose an efficient system where any authorized user can perform a search on an encrypted remote database with multiple keywords, Without revealing neither the keywords he searches for, nor the information of the documents that match with the query. The only information that the proposed scheme leaks is the access pattern which is leaked by almost all of the practical encrypted search schemes due to efficiency reasons. A typical scenario that benefits from our proposal is that a company outsources its document server to a cloud service provider. Authorized users or customers of the company can perform search operations using certain keywords on the cloud to retrieve the relevant documents. The documents may contain sensitive information about the company, and similarly the keywords a user searches for may give hints about the content of the documents, hence both must be hidden. Furthermore, search terms themselves may reveal sensitive information about the users as well, which is considered to be a privacy violation by users if learned by others.*

Keywords- Cloud computing, searchable encryption, privacy-preserving, keyword search, ranked search.

I. 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 [2], [3]. 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, for example, e-mails, personal health records, photo albums, tax documents, financial transactions, and so on, may have to be encrypted by data owners before outsourcing to the commercial public cloud [4]; 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 out sourced 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.

To enable ranked search for effective utilization of outsourced cloud data under the formation model, our system design should simultaneously achieve security and Performance guarantees as follows.

1.1 Multi-keyword ranked search:-To design search schemes which allow multi-keyword query and provide result similarity ranking for effective data retrieval, instead of returning undifferentiated results.

1.2 Privacy-preserving:-. To prevent the cloud server from learning additional information from the data set and the index, and to meet privacy requirements

1.3 Efficiency:- Above goals on functionality and privacy should be achieved with low communication and computation overhead.

II. RELATED WORK

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) [4]. It is first studied by Song et al. [7] in the symmetric key setting, and improvements and advanced security definitions are given in Goh [8], Chang

et al. [9], and Carmela et al. [10]. Our early works solve secure ranked keyword search which utilizes keyword frequency to rank results instead of returning undifferentiated results. However, they only supports single keyword search. In the public key setting, Boneh et al. [11] 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 Cipher text.

2.2 Boolean Keyword Searchable Encryption:-

To enrich search functionalities, conjunctive keyword search over encrypted data have been proposed. These schemes incur large overhead caused by their fundamental primitives, such as computation cost by bilinear map, for example, or communication cost by secret sharing, for example. As a more general search approach, predicate encryption schemes 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.

2.3 Fuzzy Keyword Searchable Encryption:-

Fuzzy keyword searches [2-4] have been developed. Chuah et al. [2] propose a privacy-aware bed-tree method to support fuzzy multi-keyword search. This approach uses edit distance to build fuzzy keyword sets. Bloom filters are constructed for every keyword. Then, it constructs the index tree for all files where each leaf node a hash value of a keyword. Li et al. [3] exploit edit distance to quantify keywords similarity and construct storage-efficient fuzzy keyword sets. Specially, the wildcard-based fuzzy set construction approach is designed to save storage overhead. Wang et al. [4] employ wildcard-based fuzzy set to build a private trie-traverse searching index. These fuzzy search methods support tolerance of minor typos and format inconsistencies, but do not support semantic fuzzy search. Considering the existence of polysemy and synonymy [5] , the model that supports multi-keyword ranked search and semantic search is more reasonable.

2.4 The Symbol-based Trie-Traverse Search Scheme:-

To enhance the search efficiency, we now propose a symbol-based trie-traverse search scheme, where a multiway tree is constructed for storing the fuzzy keyword set {S wi, d } over a finite symbol set. The key idea behind this construction is that all trapdoors sharing a common prefix may have common nodes. The root is associated with an empty set and the symbols in a trapdoor can be recovered in a search from the root to the leaf that ends the trapdoor. All fuzzy words in the trie can be found by a depth-first search. Assume $\Delta = \{a\ i\}$ is a predefined symbol set, where the number of different symbols is $|\Delta| = 2n$, that is, each symbol $a \in \Delta$ can be denoted by n bits.

2.5 Latent semantic ranked search:-

We will solve the problem of multi-keyword latent semantic ranked search over encrypted cloud data and retrieve the most relevant files. We define a new scheme named Latent Semantic Analysis (LSA)-based multi-keyword ranked search which supports multi-keyword latent semantic ranked search. By using LSA, the proposed scheme could return not only the exact matching files, but also the files including the terms latent semantically associated to the query keyword

2.5 Verifiable Search Based on Authenticated Index Structure:-

Due to possible software/hardware failure, storage corruption, etc., cloud server may return erroneous or false search results. Search result verification is a desirable feature that a robust search system would like to provide to its users. In the plaintext database scenario, verifiable search functionality has been studied extensively since the outsourced database model emerged, e.g., [15]. Similar to the works in the database, they only considered the verification-specific issues regardless of the search privacy preserving capabilities that we provide in this paper. In encrypted data search scenario, Wang et al. [9] used hash chain to construct a single keyword search result verification scheme

III. SYSTEM AND PRIVACY REQUIREMENTS

The problem that we consider is privacy-preserving keyword search on private database model, where the documents are simply encrypted with the secret keys unknown to the actual holder of the database (e.g,cloud server). We

consider three roles consistent with previous works (Wang et al., 2011):

- Data Controller is the actual entity that is responsible for the establishment of the database. The data controller collects and/or generates the information in the database and lacks the means (or is unwilling) to maintain/operate the database,
- Users are the members in a group who are entitled to access (part of) the information of the database,
- Server is a professional entity (e.g., cloud server) that offers information services to authorized users. It is often required that the server be oblivious to content of the database it maintains, the search terms in queries and documents retrieved.

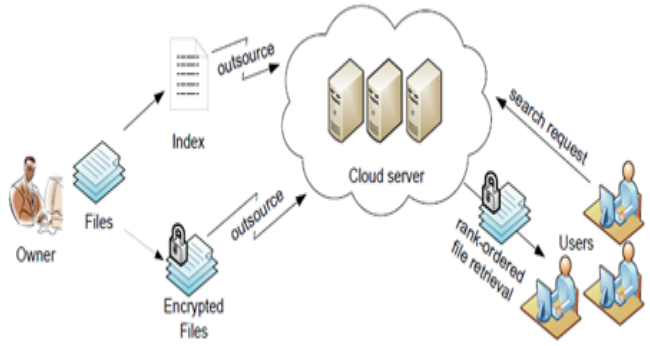

Fig 1:-Multi-keyword ranked search over encrypted cloud data Using Index Generation

Given a query from the user, the server searches over the database and returns a list of ordered items. Note that this list does not contain any useful information to the third parties. Upon receiving the list of ordered items, the user selects the most relevant data items and retrieves them.

The privacy definition for search methods in the related literature is that the server should learn nothing but the search results (i.e. access pattern) (Cao et all., 2011). We further tighten the privacy over this general privacy definition and establish a set of privacy requirements for privacy-preserving search protocols. A multi-keyword search method must provide the following user and data privacy properties (First intuitions and then formal definitions are given):

- **Query Privacy :-** The query does not leak the information of the corresponding search terms it contains.
- **Search Term Privacy:-** Given a valid query for a set of genuine search terms, no one can generate another valid query for a subset of the genuine search terms in the former query.
- **Search Pattern Privacy:-** Equality between two search requests cannot be verified by analyzing the queries or the returned list of ordered matching results.
- **Non-Impersonation:-** No one can impersonate a legitimate user.

IV. METHODOLOGY (PROPOSED SYSTEM)

The previous section introduces the three roles that we consider: Data Controller, Users and Server. Due to the privacy concern that is explained, we utilize two servers namely: index server and file Server. We assume that the parties are semi-honest (\honest but curious") and do not collude with each other to bypass the security measures, two assumptions which are consistent with most of the previous works.

In an offline stage, the data controller creates a search index element for each document. The search index file is created using a secret key based trapdoor generation function where the secret keys1 are only known by the data controller. Then, the data controller uploads this search index file to the index server and the encrypted documents to the file server. We use symmetric-key encryption as the encryption method since it can handle large document sizes efficiently. This process is referred as the index generation henceforth and the trapdoor generation is considered as one of its steps. When a user wants to perform a keyword search, he first connects to the data controller. He learns the trapdoors for the keywords he wants to search for, without revealing the keyword information to the data controller. Since the user can use the same trapdoor for many queries containing the corresponding search term, this operation does not need to be performed every time the user performs a query.

In this section, we provide the details for the crucial steps in our proposal, namely index generation, trapdoor generation, and query generation.

- **Index Generation (basic scheme):-**

Recently Wang et al., 2009 proposed a conjunctive keyword search scheme that allows multiple-keyword search in a single query. We use this scheme as the base of our index construction scheme. The original scheme uses forward indexing, which means that a searchable index file element for each document is maintained to indicate the search terms existing in the document. In the scheme of Wang et al. 2009, a

secret cryptographic hash function, that is shared between all authorized users, is used to generate the searchable index. Using a single hash function shared by several users forms a security risk since it can easily leak to the server. Once the server learns the hash function, he can break the system if the input set is small. The following example illustrates a simple attack against queries with few search terms.

Example 1 There are approximately 25000 commonly used words in English and users usually search for a single or two keywords. For such small input sets, given the hashed trapdoor for a query, it will be easy for the server to identify the queried keywords by performing a brute-force attack. For instance, assuming that there are approximately 25000 possible keywords in a database and a query submitted by a user involves two keywords, there will be $250002 < 228$ possible keyword pairs. Therefore, approximately 227 trials will be sufficient to break the system and learn the queried keywords.

- **Query Generation:-**

The search index file of the database is generated by the data controller using secret keys. A user who wants to include a search term in his query, needs the corresponding trapdoor from the data controller since he does not know the secret keys used in the index generation. Asking for the trapdoor openly would violate the privacy of the user against the data controller, therefore a technique is needed to hide the trapdoor asked by the user from the data controller.

In our proposal for obtaining trapdoors, we utilize a public hash function with uniform distribution. All keywords that exist in a document are mapped by the data controller to one of those bins using the Get Bin function. The query generation method which is summarized and works as follows. When the user connects to the data controller to obtain the trapdoor for a keyword, he first calculates the bin IDs of keywords and sends these values to the data controller. The data controller then returns the secret keys of the bins requested for, which can be used by the user to generate the trapdoors3 for all keywords in these bins. Alternatively, the data controller can send trapdoors of all keywords in corresponding bins resulting in an increase in the communication overhead. However, the latter method relieves the user of computing the trapdoors. After obtaining the trapdoors, the user can calculate the query in a similar manner to the method used by the data controller to compute the search index.

- **Oblivious Search on the Database:-**

A user's query, in fact, is just an r-bit binary sequence (independent of the number of search terms in it) and therefore, searching consists of as simple operations as binary comparison only. If the search index entry of the document (IR) has 0 for all the bits, for which the query (Q) has also 0, then the query matches to that document as shown in Equation. Note that given a query, it should be compared with search index entry of each document in the database.

- **Document Retrieval:-**

The index server returns the list of pseudo identifiers of the matching documents. If a single server is used for both search and file retrieval, it can be possible to correlate the pseudo identifiers of the matching documents and the identifiers of the actual encrypted files retrieved. Furthermore, this may also leak the search pattern that we want to hide. Therefore, we use a two-server system similar to the one proposed in [11] where the two servers are both semi-honest and do not collaborate. This method leaks the access pattern only to the file server and not to the index server, hence prevent any possible correlation between search results and encrypted documents retrieved. Subsequent to the analyze of the metadata retrieved from the index server, the user requests a set of encrypted files from the file server. The file server returns the requested encrypted files. Finally user decrypts the files and learns the actual documents. Key distribution of the document decryption keys can be done using state of the art key distribution methods and is not within the scope of this work.

In case access pattern needs also to be hidden, traditional PIR methods [2-6] or Oblivious RAM [28] can be utilized for the document retrieval process instead. However these methods are not practical even for medium sized datasets due to incurred poly logarithmic overhead.

V. CONCLUSION

The proposed solution addresses the problem of privacy-preserving ranked multi-keyword search, where the database is outsourced to remote server.. When an attacker is able to identify queries featuring the same search terms by inspecting the queries, their responses and database and search term statistics, he can mount successful attacks.

Therefore, the proposed privacy-preserving search scheme essentially implements an efficient method to satisfy query unlink ability based on query and response randomization and cryptographic techniques. Query randomization cost is negligible for data controller and even less for the user. Response randomization, on the other hand, results in a communication overhead when the response to a query is returned to the user since some fake matches are included in the response.

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] 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.

[3] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," Proc. IEEE INFOCOM, pp. 693-701, 2012.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptograpy and Data Security, Jan. 2010.

[5] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, Mar. 2001.

[6] I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishing, May 1999.

[7] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.

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

[9] Y.-C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Proc. Third Int'l Conf.Applied Cryptography and Network Security, 2005.

[10] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. 13th ACM Conf. Computer and Comm. Security (CCS 06), 2006.

[11] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2004.