Secure Access of Encrypted Cloud Data Based on Top-K Multikeywords with User Side Ranking

Sonia Ahanthem
MTech, CSE
AMC Engineering College
Bangalore, India

Bhavya Balakrishnan
Assistant professor, CSE
AMC Engineering College
Bangalore India

***Abstract*—Cloud computing has become popular due to the huge storage and flexible access from around the world. The data owners are motivated to outsource their data from local servers to the commercial public cloud for great flexibility and cost savings. But the data owner cannot trust the cloud server provider due to the confidential data. So to protect data privacy, sensitive data have to be encrypted before outsourcing to the public cloud, due to which traditional data based on plaintext keyword search is not possible. So there is a requirement for cloud data search on encrypted data. As the number of users are large and huge number of documents in the cloud, it is important to allow multiple keywords in the search request and return the resultant documents in the order of their relevance to these keywords. Existing works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely sort the search results. In this paper, for the first time, we define and solve the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). We proposed a set of strict privacy requirements for such a secure cloud data accessing system. There are many multi-keyword semantics, among those we choose the efficient similarity measure of "coordinate matching". We divide the application into three parts. First the data owner uses an encryption tool to encrypt the file to be uploaded and to generate the index file. Second, the encrypted files will be stored in the cloud, Third, The user uses a decrypted tools to decrypt the downloaded file. So this architecture assures us for better security as the used encryption schemes will not exposed to the network and the cloud. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given. Experiments on the real-world data set further show proposed schemes indeed introduce low overhead on computation and communication.**

Key Words —Cloud computing, searchable encryption, privacy-preserving, keyword search, ranked search.

I. INTRODUCTION

CLOUD computing emerging technology becoming popular day by day, where cloud customers can remotely store their data into the cloud so as to enjoy the on-demand high-quality applications and services and stored data from cloud. Its great flexibility and economic savings are motivating both individuals and enterprises to outsource their local data into the cloud. To protect data privacy and easy and secured accesses in the cloud, data owners upload and stores data into the cloud. To access the data stored by the owner, users need to search by keywords. 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 a challenging job. 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.

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 [5]. 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-you-use" cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information. 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 may tend to provide a set of keywords instead of only one user 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" [6], 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.

In the literature, searchable encryption [7], [8], [9], 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. Our early works, have been aware of this problem, and provide solutions to the secure ranked search over encrypted data problem but only for queries consisting of a single keyword. How to design an efficient encrypted data search mechanism that supports multi-keyword semantics without privacy breaches still remains a challenging open problem.

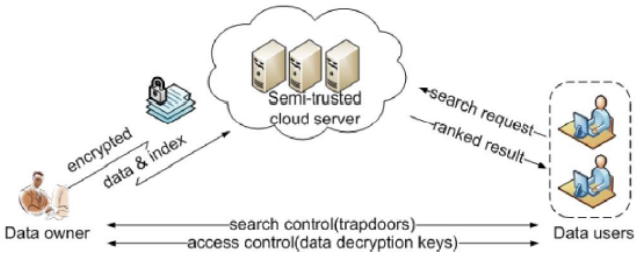

Fig 1. Architecture

In this proposal, for the first time, we define and solve the problem of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving strict system wise privacy in the cloud computing paradigm. Among various multi-keyword semantics, we choose the efficient similarity measure of "coordinate matching," i.e., as many matches as possible, to capture the relevance of data documents to the search query. During the index construction, each document is associated with an index file to check whether corresponding keyword is contained in the document. To achieve this the data owner while uploading the files generates the index files containing the important keywords contained by the file he is uploading. Data owner encrypts the file using AES algorithms that needs a key. Data owner uploads the encrypted file along with the index file. User while searching for files of his/her needs sends multi keywords to server. Cloud server checks the keywords in the index files instead of searching in the document sets. By doing this it takes very less time to find the relevant documents. After the searching process, cloud server gets the related document corresponding to the keywords in the index file. Once the user gets the list of documents with a little description ranks the files and sends back to the cloud. Cloud server after receiving the response from the user sends back the interested files to the user. User after receiving the files in encrypted format asks for the key to the data owner and then decrypt the files. In such an architecture the cloud server cannot come to know about the encryption techniques.
Our overall architecture is described in the fig 2.
Compared with the preliminary version, this paper version proposes two new mechanisms to support more search semantics. This version also studies the support of data/index dynamics in the mechanism design. Moreover, we improve the experimental works by adding the analysis and evaluation of two new schemes. In addition to these improvements, we add more analysis on secure inner product and the privacy part.

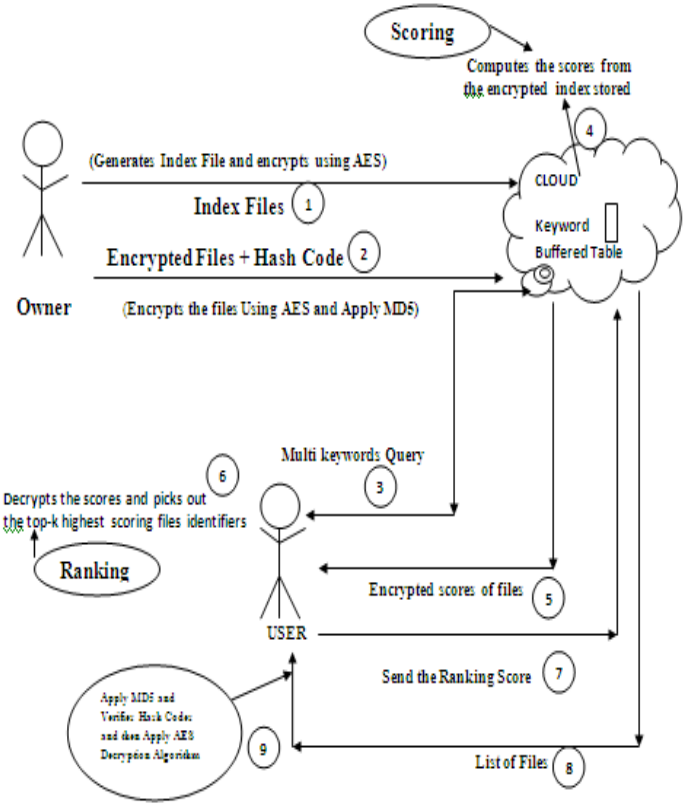

Fig 2. Working Diagram

II. SYSTEM ANALYSIS

A. *System Model*

Observe a cloud data hosting service involving three different entities, as illustrated in Fig. 1: the data owner, the data user, and the cloud server. The data owner has a collection of data documents F to be outsourced to the cloud server in the encrypted form C. To enable the searching capability over C for effective data utilization, the data owner, before outsourcing, will first build an encrypted searchable index I from F, and then outsource both the index I and the encrypted document collection C to the cloud server. To search the document collection for t given keywords, an authorized user acquires a corresponding trapdoor T through search control mechanisms, for example, broadcast encryption [10]. Upon receiving T from a data user, the cloud server is responsible to search the index I and return the corresponding set of encrypted documents. To improve the document retrieval accuracy, the search result should be ranked by the cloud server according to some ranking criteria (e.g., coordinate matching, as will be introduced shortly). Moreover, to reduce the communication cost, the data user may send an optional number k along with the trapdoor T so that the cloud server only sends back top-k documents that are most relevant to the search query. Finally, the access control mechanism [28] is employed to manage decryption capabilities given to users and the data collection can be updated in terms of inserting new documents, updating existing documents, and deleting existing documents.

B. Threat Model

The cloud server is considered as "honest-but-curious" in our model, which is consistent with related works on cloud security [28], [29]. Specifically, the cloud server acts in an "honest" fashion and correctly follows the designated protocol specification. However, it is "curious" to infer and analyze data (including index) in its storage and message flows received during the protocol so as to learn additional information. Based on what information the cloud server knows, we consider two threat models with different attack capabilities as follows.

Known cipher text model. In this model, the cloud server is supposed to only know encrypted data set C and searchable index I, both of which are outsourced from the data owner.

Known background model. In this stronger model, the cloud server is supposed to possess more knowledge than what can be accessed in the known cipher text model. Such information may include the correlation relationship of given search requests (trapdoors), as well as the data set related statistical information. As an instance of possible attacks in this case, the cloud server could use the known trapdoor information combined with document/keyword frequency to deduce/identify certain keywords in the query.

III. PRIVACY REQUIREMENTS FOR MRSE

The framework of multi-keyword ranked search over encrypted cloud data (MRSE) and establish various strict system wise privacy requirements for such a secure cloud data utilization system are defined.

AES-256:-

AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits. AES operates on a 4×4 column-major order matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.

MD5: The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly.

The MD5 algorithm is an extension of the MD4 message-digest algorithm [1.2]. MD5 is slightly slower than MD4, but is more "conservative" in design. MD5 was designed because it was felt that MD4 was perhaps being adopted for use more quickly than justified by the existing critical review; because MD4 was designed to be exceptionally fast, it is "at the edge" in terms of risking successful cryptanalytic attack. MD5 backs off a bit, giving up a little in speed for a much greater likelihood of ultimate security. It incorporates some suggestions made by various reviewers, and contains additional optimizations. The MD5 algorithm is being placed in the public domain for review and possible adoption as a standard.

A. Requirements

In the proposed system we have introduced a Keyword Buffered Controller in the server. On receiving a request from a user, the Keyword Buffered Controller stores the keywords in the Keyword Buffered Table and on retrieval of the relative files; it stores the file names in the Keyword Buffered Table along with the keywords. When the next request from a user, Keyword Buffered Controller first check in the Keyword Buffered Table, if it finds the matching encrypted keywords in the table, it fetch out the related file names and sends the content to the user. By doing so the computational task of the cloud is reduced.

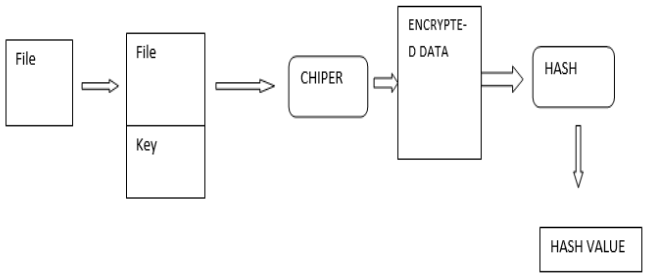

Fig 3. Encoding Operation

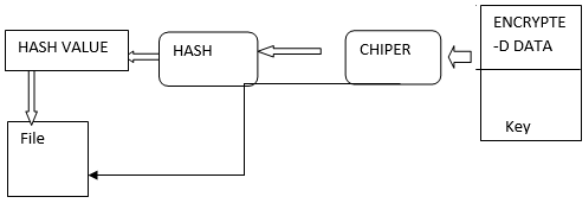

Fig 4. Restoring Data

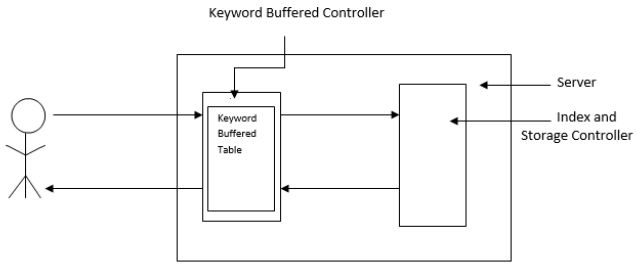

Fig 5. Keyword Buffered Controller

IV. PERFORMANCE ANALYSIS

A. *Precision & Privacy*

Dummy keywords are inserted into each data vector and some of them are selected in every query. Therefore, similarity scores of documents will be not exactly accurate. In other words, when the cloud server returns top-k documents based on similarity scores of data vectors to query vector, some of real top-k relevant documents for the query may be excluded. This is because either their original similarity scores are decreased or the similarity scores of some documents out of the real top-k are increased, both of which are due to the impact of dummy keywords inserted into data vectors. To evaluate the purity of the k documents retrieved by user, we define a measure as precision Pk ¼ k0=k where k0 is number of real top-k documents that are returned by the cloud server.

B. *Efficiency*

To build a searchable sub index Ii for each document Fi in the data set F, the first step is to map the keyword set extracted from the document Fi to a data vector Di, followed by encrypting every data vector. The time cost of mapping or encrypting depends directly on the dimensionality of data vector which is determined by the size of the dictionary, i.e., the number of indexed keywords. And the time cost of building the whole index is also related to the number of sub index which is equal to the number of documents in the data set. Given the same dictionary where the time cost of building the whole index is nearly linear with the size of data set since the time cost of building each subindex is fixed.

C. *Query*

Query execution in the cloud server consists of computing and ranking similarity scores for all documents in the data set. The two schemes in the known cipher text models have very similar query speed since they have the same dimensionality which is the major factor deciding the computation cost in the query. The query speed difference also caused by the dimensionality of data vector and query vector. With respect to the communication cost in Query, the size of the trapdoor is the same as that of the sub index listed, which keeps constant given the same dictionary, no matter how many keywords are contained in a query. While the computation and communication cost in the query procedure is linear with the number of query keywords in other multiple-keyword search schemes our proposed schemes introduce nearly constant overhead while increasing the number of query keywords. Therefore, our schemes cannot be compromised by timing-based side channel attacks that try to differentiate certain queries based on their query time.

D. *Single keyword searchable encryption*

Traditional single keyword searchable encryption 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 [9], and Curtmola [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.

E. *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, or communication cost by secret sharing. 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. 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. Our early work [1] has been aware of this problem, and provides solutions to the multi-keyword ranked search over encrypted data problem. In this paper, we extend and improve more technical details as compared to [1]. We propose two new schemes to support more search semantics which improve

the search experience of the MRSE scheme, and also study the dynamic operation on the data set and index which addresses some important yet practical considerations for the MRSE design. On a different front, the research on top-k retrieval in database community is also loosely connected to our problem. Besides, Cao proposed a privacy-preserving graph containment query scheme which solves the search problem with graph semantics.

V. CONCLUSIONS

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, choose the efficient similarity measure of "coordinate matching," i.e., as many matches as possible, to effectively capture the relevance to the query keywords, and use "inner product similarity" to quantitatively evaluate such similarity measure. For meeting the challenges of supporting multi-keyword semantic without privacy breaches, we put forward a basic idea of MRSE using secure inner product computation. Then, we give two improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. We also investigate some further enhancements of our ranked search mechanism, including supporting more search semantics, i.e., TF & IDF, and dynamic data operations. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given, and experiments on the real-world data set show our proposed schemes introduce low overhead on both computation and communication.

VI. REFERENCES

[1] N. Cao, C Wang, Z. Li, K Ren and W. Lou, "Privacy Preserving multikeyword ranked search over encrypted cloud data," Proc. IEEE INFOCOM, PP. 829-837,2012.

[2] L.M. Vaquero, L. Rodero-Merino, Caceres, and M Lindner, "A break in the clouds: towards a cloud definition," AC compute. Communes. 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.