Multi-keyword Stratified Search over Encrypted Cloud Data

P. Uma Rani[1], Dr. B. Jhansi Vazram[2], G. Raphi[3]

[1] *M.Tech Student, Department of CSE, Narasaraopet Engineering College, Narasaraopet*
Guntur dist, A.P, India
[2] *Professor, NEC, Narasaraopet, Guntur dist, A.P, India*
[3] *Assistant Professor, NEC, Narasaraopet*
Guntur dist, A.P, India
[1] umapadyala@gmail.com
[2] jhansi.bolla@gmail.com
[3] rafi1222@gmail.com

***Abstract*—The advancement in cloud computing has motivated the data owners 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 have to be encrypted before outsourcing, which antiquates traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of vital importance. Considering the large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely sort the search results. In this paper, we define and solve the challenging problem of Multi-keyword Stratified Search over Encrypted cloud data (MSSE). We propose a basic idea for the MSSE based on secure inner product computation. Experiments on the real-world data set further show proposed scheme indeed introduce low overhead on computation and communication.**

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. 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. 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.

Ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection .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.

In this paper, we define and solve the problem of multi-keyword stratified search over encrypted cloud data (MSSE), 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", to capture the relevance of data documents to the search query. Specifically, we use "inner product similarity" to quantitatively evaluate such similarity measure of that document to the search query. During the index construction, each document is associated with a binary vector as a sub index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by the inner product of the query vector with the data vector. However, directly outsourcing the data vector or the query vector will violate the index privacy or the search privacy.

To meet the challenge of supporting such multi keyword semantic without privacy breaches, we propose a basic idea for the MSSE using secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique to achieve various stringent privacy requirements. Our contributions are summarized as follows:

- For the first time, we explore the problem of multi keyword ranked search over encrypted cloud data, and establish a set of strict privacy requirements for such a secure cloud data utilization system.
- We investigate some further enhancements of our ranked search mechanism to support more search semantics and dynamic data operations.
- Thorough analysis investigating privacy and efficiency guarantees of the proposed scheme is given, and experiments on the real-world data set further show the proposed scheme indeed introduce low overhead on computation and communication.

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 keys. 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, 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. 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 has been aware of this problem, and provides solutions to the multi-keyword ranked search over encrypted data problem.

III. FRAME WORK

Consider a cloud environment with three different entities:

- Data Owners
- Data Users
- Semi-trusted Cloud server

The architecture of the search over encrypted cloud data was shown in Fig 1. As per our frame work the data owner will have all the files like doc, pdf, txt...etc. The data owner is going to encrypt all those files before outsourcing with some encryption keys. The encrypted files are going to be stored in the semi-trusted cloud server, at the same time he creates an encrypted searchable index. Then he is going to outsource those collections of encrypted files into semi-trusted cloud server. To search the documents, data user acquires a corresponding trapdoor. Upon receiving trapdoor from a data user, the cloud server is responsible to search the index and return the corresponding set of encrypted documents.

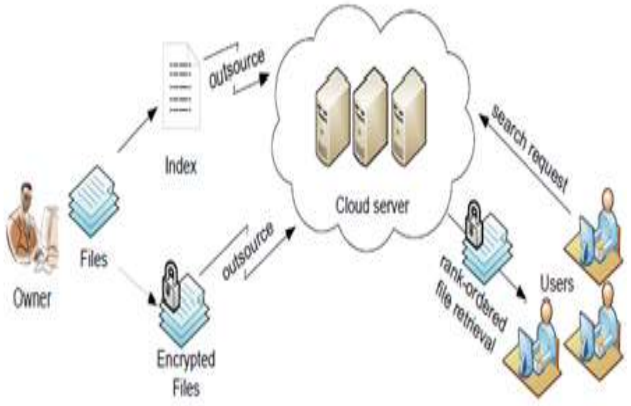

Fig.1. Architecture of the search over encrypted cloud data.

- **Data Owner**: He is going to upload his data what he want to share using cloud. Before out sourcing he encrypts all the files and provides index terms for his data.
- **Data User**: Data user searches the required files on the encrypted cloud data.
- **Semi-trusted cloud server:** Here the data owners can store their encrypted files. Users can search and get the required data using index terms.

Previously user can search only plain text formatted files and he is not able to search different types of text formats like doc, pdf...etc. Because in the existing system, there is no such mechanism to search different types of text files. And there is no such decryption technique to represent all types of encrypted files. Our proposed system uses one technique, using which we can directly search all types of files and it is encryption based searching technique. This technique provides efficient solution for accessing different types of data.

IV. MSSE FRAME WORK

Multi-keyword stratified search schemes allow multi-keyword query and provide result similarity ranking for effective data retrieval, instead of returning undifferentiated results.

Here we define the framework of multi-keyword stratified search over encrypted cloud data (MSSE) and establish various strict system wise privacy requirements for such a secure cloud data utilization system. The operations on the data documents are not shown within the framework. Since the data owner may simply use the traditional symmetric key cryptography to encrypt and then outsource data. With focus on the index and query, the MSSE system consists of the following:

- **Setup**: Taking a security parameter l as input, the data owner outputs a symmetric key as SK.
- **Build Index**: Based on the data set F, the data owner builds a searchable index I which is encrypted by the symmetric key SK and then outsourced to the cloud server. After the index construction, the document collection can be independently encrypted and outsourced.

- **Trapdoor**: With t keywords of interest in W as input, this algorithm generates a corresponding trapdoor.
- **Query**: When the cloud server receives a query request, it performs the ranked search on the index I with the help of trapdoor, and finally returns the ranked id list of top-k documents sorted by their similarity with W.

4.1 Notations

- F—the plaintext document collection, denoted as a set of m data documents F = (F_1, F_2,…, F_m).
- C —the encrypted document collection stored in the cloud server, denoted as C = (C_1, C_2,…, C_m).
- W—the dictionary, i.e., the keyword set consisting of *n* keyword, denoted as W= (W_1, W_2,…, W_n).
- I—the searchable index associated with C, denoted as (I_1, I_2,…, I_m).
- T_w —the trapdoor for the search request.
- F_w— the ranked id list of all documents according to their relevance.

4.2 Privacy Requirements for MSSE

We explore and establish a set of strict privacy requirements specifically for the MSSE framework.

As for the data privacy, the data owner can resort to the traditional symmetric key cryptography to encrypt the data before outsourcing, and successfully prevent the cloud server from prying into the outsourced data. With respect to the index privacy, if the cloud server deduces any association between keywords and encrypted documents from index, it may learn the major subject of a document, even the content of a short document. Therefore, the searchable index should be constructed to prevent the cloud server from performing such kind of association attack. While data and index privacy guarantees are demanded by default in the related literature, various search privacy requirements involved in the query procedure are more complex and difficult to tackle as follows:

4.2.1 Keyword Privacy:

As users usually prefer to keep their search from being exposed to others like the cloud server, the most important concern is to hide what they are searching, i.e., the keywords indicated by the corresponding trapdoor. Although the trapdoor can be generated in a cryptographic way to protect the query keywords, the cloud server could do some statistical analysis over the search result to make an estimate. As a kind of statistical information, document frequency (i.e., the number of documents containing the keyword) is sufficient to identify the keyword with high probability.

4.2.2 Trapdoor Unlinkability:

The trapdoor generation function should be a randomized one instead of being deterministic. In particular, the cloud server should not be able to deduce the relationship of any given trapdoors, for example, to determine whether the two trapdoors are formed by the same search request. Otherwise, the deterministic trapdoor generation would give the cloud server advantage to accumulate frequencies of different search requests regarding different keyword(s), which may further violate the aforementioned keyword privacy requirement. So the fundamental protection for trapdoor unlinkability is to introduce sufficient non-determinacy into the trapdoor generation procedure.

4.2.3 Access Pattern:

Within the ranked search, the access pattern is the sequence of search results where every search result is a set of documents with rank order. Specifically, the search result for the query keyword set W is denoted as F_W, consisting of the id list of all documents ranked by their relevance to F_W. Then the access pattern is denoted as (F_{W1}, F_{W2} ...) which are the results of sequential searches.

4.2.4 *MSSE Scheme*

To efficiently achieve multi-keyword stratified search, we propose to employ “inner product similarity” to quantitatively evaluate the efficient similarity measure “coordinate matching.” Specifically, D_i is a binary data vector for document F_i where each bit $D_i[j]$ € {0, 1} represents the existence of the corresponding keyword W_j in that document, and Q is a binary query vector indicating the keywords of interest where each bit Q[j] € {0, 1} represents the existence of the corresponding keyword W_j in the query W. The similarity score of document F_i to query W is therefore expressed as the inner product of their binary column vectors, i.e., D_i. Q. For the purpose of ranking, the cloud server must be given the capability to compare the similarity of different documents to the query. But, to preserve strict system wise privacy, data vector D_i, query vector Q and their inner product D_i. Q should not be exposed to the cloud server. In this section, we propose MSSE frame work in a step-by-step manner.

In our more advanced design, instead of simply removing the extended dimension in the query vector as we plan to do at the first glance, we preserve this dimension extending operation but assign a new random number t to the extended dimension in each query vector. Such a newly added randomness is expected to increase the difficulty for the cloud server to learn the relationship among the received trapdoors. In addition, as mentioned in the keyword privacy requirement, randomness should also be carefully calibrated in the search result to obfuscate the document frequency and diminish the chances for re identification of keywords. Introducing some randomness in the final similarity score is an effective way toward what we expect here. The whole scheme to achieve ranked search with multiple keywords over encrypted data is as follows:

- **Setup**: The data owner randomly generates a bit vectors S and two invertible matrices {M1, M2}. The secret key SK is in the form of a 3-tuple as {S, M1, M2}.
- **Build Index (F,SK):** The data owner generates a binary data vector D_i for every document F_i, where each binary bit represents whether the corresponding keyword W_j appears in the document $F_{i.}$
- **Trapdoor:** With t keywords of interest in W as input, one binary vector Q is generated where each bit indicates whether Wj € W is true or false. After applying the same splitting and encrypting processes, the trapdoor is generated.

- **Query:** With the trapdoor $t_{w,}$, the cloud server computes the similarity scores of each document F_i. After sorting all scores, the cloud server returns the top-k ranked id list $F_{w.}$

V. PERFORMANCE ANALYSIS

In this section, we demonstrate a thorough experimental evaluation of the proposed technique on a real-world data set.

5.1 Efficiency

5.1.1 Index Construction

To build a searchable sub index I_i for each document F_i in the data set F, the first step is to map the keyword set extracted from the document Fi to a data vector D_i, 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. Fig. 2 shows that, given the same dictionary where W=4,000, the time cost of building the whole index is nearly linear with the size of data set since the time cost of building each sub index is fixed. The size of sub index is absolutely linear with the dimensionality of data vector which is determined by the number of keywords in the dictionary.

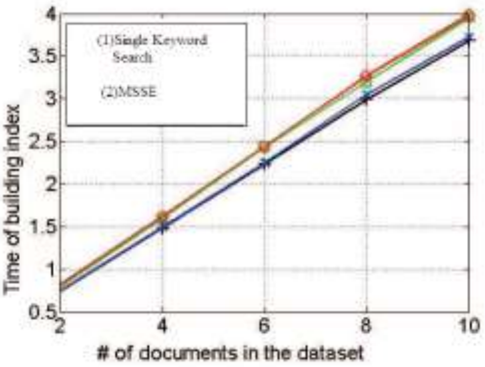

Fig.2. Time cost of building index.

VI. CONCLUSION

In this paper, for the first time we define and solve the Problem of multi-keyword stratified 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. For meeting the challenge of supporting multi-keyword semantic without privacy breaches, we propose a basic idea of MSSE using secure inner product computation. 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.

5.2.2 Query

Query execution in the cloud server consists of computing and ranking similarity scores for all documents in the data set. The computation of similarity scores for the whole data collection is O(mn). Fig.3. shows the query time is dominated by the number of documents in the data set. The computation and communication cost in the query procedure is linear with the number o 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.

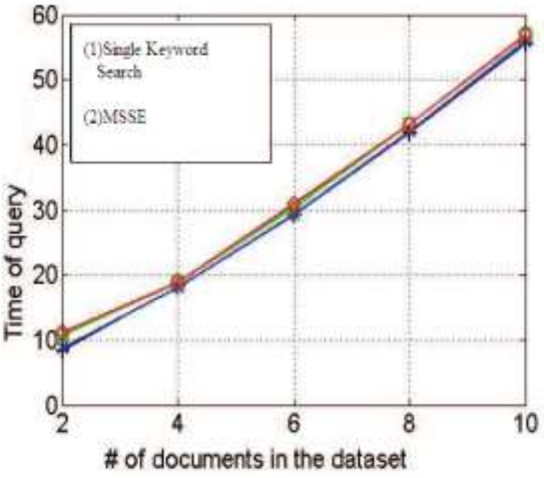

Fig.3. Time cost of query

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 INFO-COM, 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 SIGOMM 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] 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*.

[9] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key of Encryption with Keyword Search,” *Proc. Int’l Conf. Theory and applications of Cryptographic Techniques (EUROCRYPT),* 2004.

[10] M. Bellare, A. Boldyreva, and A. ONeill, “Deterministic and Efficiently Searchable Encryption,” *Proc. 27th Ann. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO ’07),* 2007.

[11] 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*.

[12] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data, " Proc. Fourth Conf. Theory Cryptography (TCC), pp. 535- 554, 2007.

[13] R. Brinkman, "Searching in Encrypted Data," PhD thesis, Univ. of Twente, 2007.

[14] Y. Hwang and P. Lee, "Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-User System," Pairing, vol. 4575, pp. 2-22, 2007.

[15] J. Katz, A. Sahai, and B. Waters, "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products*," Proc. 27th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2008.*

Selected Paper from International Conference on Computing (NECICC-2k15)