A Secure Greedy Depth First Search Algorithm for Encrypted Data in Cloud Computing Environment

Yesaswini K
M.Tech Scholar
Department of Computer Science & Engineering
Don Bosco Institute of Technology

Dr. S. Meenakshi Sundaram
HOD & Professor
Department of Computer Science & Engineering
Don Bosco Institute of Technology

***Abstract*— Nowadays, more and more people are forced to outsource their local data to public cloud servers for great convenience and reduced expenses in data management. But in consideration of privacy issues, sensitive data should be encrypted before outsourcing, which obsoletes usual data utilization like keyword-based document retrieval. In this paper, we present a protected and well-organized multi-keyword ranked search scheme over encrypted data, which additionally supports active update operations like removal and addition of documents. Specifically, we construct an index tree based on vector space sculpt to provide multi-keyword search, which meanwhile supports flexible update operations. Besides, cosine similarity measure is utilized to support exact ranking for search result. To improve search efficiency, we further develop a search algorithm based on Greedy Depth-first Search Strategy. Moreover, to protect the search privacy, we propose a protected scheme to meet various confidentiality requirements in the known ciphertext threat model. Experiments on the real-word dataset show the success and competence of proposed scheme.**

Keywords*— *Cloud Computing, Depth first search, Multi keyword search, Encrypted data

I. INTRODUCTION

Cloud Computing enables cloud customers to distantly store their data into the cloud so as to enjoy the on-demand high quality applications and armed forces from a common pool of configurable computing resources [1]. The benefits brought by this new computing model include but are not restricted to relief of the burden for storage management, universal data entrance with autonomous geographical locations, and avoidance of capital expenditure on hardware, software, and natives management, etc [2]. Cloud computing is a term used to describe a set of IT services that are provided to a buyer over a network on a rent basis and with the ability to scale up or down their service requirements. Clouds are large pools of easily usable and easy to get to virtualized resources. These resources can be dynamically reconfigured to adjust to a uneven load (scale), allowing optimum resource utilization. It‟s a pay-per-use model in which the Infrastructure contributor by means of customized Service Level Agreements (SLAs)[1] offers guarantees typically exploiting a group of resources. Organizations and individuals can advantage from mass computing and storage centers, provided by great companies with constant and well-built cloud architectures.

Latest years, cloud computing enjoys great status in data management due to its excellent potential in computing, storage and a variety of applications. From beginning to end cloud services, people could enjoy convenient, on-demand network access to a common group of configurable computing assets with great efficiency and minimal economic overhead [1]. Despite of the various benefits on hand by cloud services, shift of sensitive information (such as e-mails, company finance data, and administration credentials, etc) to semi-trusted cloud server brings concerns about privacy problems. For instance, the cloud server may leak information to legitimate entities or even be hacked, which creates the outsourced data at risk. Conventionally, susceptible data should be encrypted by data proprietor before outsourcing, which, however, obsoletes predictable data consumption service like keyword-based information retrieval.

To enable conventional utilization on encrypted data, searchable encryption techniques [2-11], especially those based on symmetric key cryptography [4-11], are proposed for well-organized keyword search over encrypted data. Before now, many symmetric searchable encryption (SSE) schemes have been developed as an effort for enriching the search elasticity, like single-keyword search [4-10] and multi-keyword search [11-11]. To enhance the accuracy of search result, multi-keyword search is well appropriate for real world than single-keyword search. Along with those multi-keyword search mechanism, many have realized the conjunctive keyword search, subnet search, or range search [10, 11], but they don’t support exact ranked search. In plaintext information retrieval (IR) community, there are many up to date technologies for multi-keyword ranked search, for instance, cosine measure in the vector space model [11]. To provide ranked search efficiency on encrypted data, Cao et al. [11] proposed a privacy-preserving multi-keyword ranked search scheme. With synchronize matching, search result is ranked according to the number of matched keywords, which is not perfect enough. And their search complexity is linear with the number of documents in dataset. Then, in [10], Sun et al. proposed a multi-keyword search scheme that supports similarity-based ranking. They constructed a searchable directory tree based on vector space sculpt and adopted cosine measure together with term frequency (TF) × inverse document frequency (IDF)to provide correct ranking. Finally, their search algorithm

achieves better-than-linear search efficiency. Besides, in Cloud Computing, information owners may share their outsourced information with a large number of clients, who might want to only regain certain explicit information files they are interested in during a given assembly. One of the most traditional ways to do so is through keyword-based search. Such keyword search method allows users to selectively take files of interest and has been widely applied in plaintext search states. Unfortunately, information encryption, which restricts customer's ability to perform keyword search and further demands the protection of keyword privacy, makes the typical plaintext search methods fail for encrypted cloud data.

This paper focuses on to the explanation of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving firm system-wise privacy in the cloud computing model. A variety of multi- keyword semantics are available, an well-organized comparison measure of coordinate matching, i.e., as several matches as possible, to capture the relevance of data documents to the search query is used.

II. RELATED STUDY

Moreover, in Cloud Computing,owners may contribute their outsourced information with a large number of customers, who might want to only retrieve certain specific information files they are concerned in during a given gathering. One of the most established habits to do so is through keyword-based search. Such keyword search method allows users to selectively get back files of concern and has been widely functional in plaintext search states. Unfortunately, information encryption, which restrict customer's competence to perform keyword search and further demands the protection of keyword privacy, makes the common plaintext search methods not succeed for encrypted cloud data.

Cloud computing is a term used to describe a set of IT services that are given to a customer over a network on a leased basis and with the ability to level up or down their service necessities. Clouds are great pools of easily usable and available virtualized resources. These resources can be vigorously reconfigured to adjust to a uneven load (scale), allowing optimum resource utilization. It‟s a pay-per-use model in which the Infrastructure Provider by means of customized Service Level Agreements (SLAs)[1] offers guarantees typically exploiting a group of resources. Organizations and individuals can benefit from mass computing and storage centers, given by huge companies with constant and well-built cloud architectures.

The encrypted data to the cloud and perform keyword search over ciphertext domain. Due to different cryptography Primitives, searchable encryption strategy can be constructed using public key based cryptography. or symmetric key based cryptography. Song *et al.* proposed the first symmetric searchable encryption (SSE) scheme, and the look for time of their scheme is linear to the amount of the data gathering. Goh [8] proposed prescribed security definitions for SSE and considered a scheme based on Bloom filter. The search time of Goh's scheme is $O(n)$, where n is the cardinality of the document gathering. Curtmola *et al.* [10] anticipated two schemes (SSE-1 and SSE-2) which achieve the optimal search time. Their SSE-1 scheme is secure against selected-keyword attacks (CKA1) and SSE-2 is secure against adaptive chosen-keyword attacks (CKA2). These early works are single keyword boolean search schemes, which are very simple in terms of functionality. Afterward, abundant works have been proposed under different danger models to achieve various search functionality, such as single keyword search, similarity ,multi-keyword boolean search, ranked search, and multi-keyword ranked search etc.

Multiple query keywords to request suitable documents. Among these works, conjunctive keyword search schemes only return the documents that contain all of the query keywords. Disjunctive keyword search schemes displays all of the documents that contain a subset of the query keywords. Predicate search schemes are proposed to support both conjunctive and disjunctive search. All these multikeyword search schemes get back search results based on the existence of keywords, which cannot provide acceptable outcome ranking functionality. Ranked search can enable quick search of the most relevant data. Transferring only the top-k most relevant documents can effectively reduces network traffic. Some early works have realized the ranked search using order-preserving techniques, but they are designed only for single keyword search. Cao *et al.* realized the first privacy-preserving multi-keyword ranked search scheme, in which documents and queries are represented as vectors of dictionary size. With the "coordinate matching", the documents are ranked according to the number of matched query keywords. However, Cao *et al.*'s scheme does not consider the importance of the different keywords, and thus is not very much correct. In addition, the search efficiency of the scheme is linear with the cardinality of document gathering.

Sun *et al.* presented a secure multi-keyword search scheme that supports comparison-based ranking. The authors constructed a searchable index tree based on vector space model and adopted cosine measure together with TF×IDF to provide ranking results. Sun *et al.*'s search algorithm achieves better-than-linear search efficiency but outcomes in accuracy loss. O¨ rencik *et al.* proposed a secure multi-keyword search method which utilized local sensitive hash (LSH) functions to cluster the similar documents. The LSH algorithm is suitable for comparable search but cannot provide correct ranking. In , Zhang *et al.* proposed a scheme to deal with secure multi-keyword ranked search in a multi-owner model. In this scheme, different data owners use different secret keys to encrypt their documents and keywords while authorized data users can query without perceptive keys of these different data owners. The authors proposed an "Additive Order Preserving Function" to retrieve the most appropriate search results. However, these works don't support dynamic operations.

In 2012, Kamara et al. [30] constructed an encrypted inverted index that can switch active data efficiently But, this scheme is very difficult to execute. Subsequently, as an improvement, Kamara and Papamanthou [31] proposed a new search scheme based on tree-based index, which can switch active update on document data stored in leaf nodes.

However, their scheme is considered only for single-keyword Boolean search. In [09], Cash et al. presented a data structure for keyword/identity tuple named T-Set. Then, a document can be represented by a series of independent T-Sets. Based on this structure, Cash et al. [09] proposed a dynamic searchable encryption scheme. In their construction, newly added tuples are stored in one more database in the cloud, and deleted tuples are stored in a revocation list. The final search outcome is achieved through apart from tuples in the revocation list from the ones retrieved from original and newly added tuples. Yet, Cash et al. active search scheme doesn't understand the multi-keyword ranked search importances.

III. PROBLEM STATEMENT

Sensitive cloud information have to be encrypted to care for information security, before outsourced to the marketable public cloud. The encryption process makes successful information utilization service a very difficult job. Traditional searchable encryption techniques allow users to steadily search over encrypted information through keywords Searchable encryption technique wires only Boolean search process. Large amount of users and information files are not efficiently managed by the searchable encryption sculpt. The privacy enabled information searching scheme gives way for protected ranked keyword search over encrypted cloud data. Ranked search improves system usability by providing search result significance ranking.

Relevance score is a numerical calculation approach is used in information retrieval. Relevance score is used in protected searchable directory construction process. One-to-many order-preserving mapping technique is used to appropriately secure those sensitive score information. The system provides server-side ranking without losing keyword privacy. Ranked Searchable Symmetric Encryption (RSSE) scheme is used to accomplish secured data retrieval process.

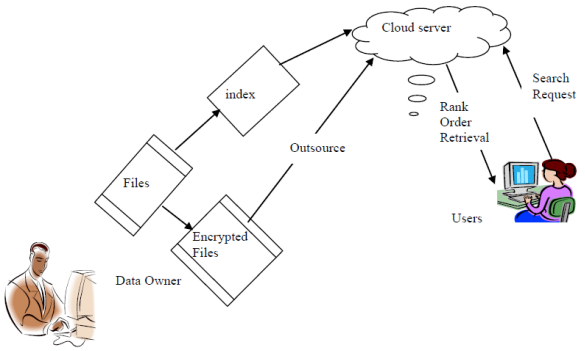

Figure 1: Search Over encrypted Cloud Data Architecture

A. System Model

Cloud data storage service has three actors 1. Data owner, Data User and Cloud Server as shown in Fig. 1. Data owners stores a set of document D on to the cloud server in an encrypted form to keep away from the security problems. To enable fast and cost effective data retrieval a search index I is built over an encrypted data. To make a search, a set of keywords K is given by an approved user. The results are ranked using the ranking algorithm by the cloud server.

B. Design Objective

To activate the ranked search for effective utilization of outsourced cloud data under the above mentioned model, the system should be designed by considering the security considerations also. The system is expected to give the following security and performance guarantees as follows.

- Multi-keyword Ranked Search: To propose search schemes which allow multi-keyword query and gives outcomes comparison ranking for effective data retrieval, instead of returning undifferentiated results.
- Privacy-Preserving: To prevent the cloud server from learning extra information from the document collection and the index, and to satisfy the essential privacy requirements.
- Efficiency: Ranked search should guarantee confidentiality and also low communication and estimation overhead.

IV. PROPOSED METHODOLOGY

For easy appearance, procedures on the data documents are not shown in the framework since the data owner could easily make use of the well organised symmetric key cryptography to encrypt and then outsource data. With focus on the index and query, the MRSE system has of four algorithms as follows:

1. ***Setup(ℓ)***

Taking a security parameter ℓ as input, the data owner displays a symmetric key as SK.

2. ***BuildIndex(F, SK)***

Based on the dataset 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 gathering can be separately encrypted and outsourced.

3. ***Trapdoor(fW)***

With t keywords of interest in fW as input, this algorithm generates a equivalent trapdoor TfW.

4. ***Query(TfW, k, I)***

When the cloud server fetches a query request as (TfW, k), it performs the ranked search on the index I with the help of trapdoor TfW, and at the end returns FfW, the ranked id gives of top-k documents sorted by their similarity with fW.

A. Security and Privacy Requirements for MRSE Framework

The delegate privacy guarantee in the related literature, such as searchable encryption, is that the server should learn nothing but search outcomes. With this general privacy description, we discover and set up a set of strict privacy requirements specifically for the MRSE framework. As for the data privacy, the data owner can route to the traditional symmetric key cryptography to encrypt the data before outsourcing, and successfully prevent the cloud server from interfering into the outsourced data. With respect to the index privacy, [9][10] if the cloud server produces any relationship 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.

Keyword Privacy:

As users usually wish to keep their search from being showing to others like the cloud server, the most important apprehension is to secrete what they are searching, i.e., the keywords indicated by the corresponding trapdoor. Although the trapdoor can be generated in a binary format way to protect the query keywords, the cloud server could do some numerical examination over the search result to make an guess [11]. As a kind of statistical information, *document frequency* (i.e., the number of documents having the keyword) is sufficient to identify the keyword with high probability. When the cloud server knows some background information of the samples, this keyword explicit data may be utilized to reverse-engineer the keyword.

Trapdoor Unlinkability: The trapdoor creation [5][6] function should be a randomized instead of being deterministic. In particular, the cloud server should not be able to derive the association of any given trapdoors, e.g., to determine whether the two trapdoors are formed by the same search request. Otherwise, the deterministic trapdoor creation would give the cloud server advantage to build up frequencies of different search requests regarding different keyword(s), which may further violate the above mentioned keyword privacy requirement. So the fundamental protection for trapdoor unlinkability is to introduce adequate and no determinacy into the trapdoor generation procedure.

Access Pattern: Within the ranked search, the access example is the sequence of search results where every search result is a set of documents with rank order. Particularly, the search result for the query keyword set fW is denoted as FfW, consisting of the id directory of all documents ranked by their relevance to fW. Then the access pattern is denoted as (FfW1 ,FfW2 , . . .) which are the results of sequential searches. Although a few searchable encryption works, e.g., [11] has been proposed to make use of private information retrieval (PIR) technique [28], to secure the access examples, our proposed schemes are not designed to secure the access examples for the competence concerns. This is because any PIR based technique must "touch" the whole samples outsourced on the server which is incompetent in the large size cloud system.

In our more sophisticated design, instead of simply deleting the extended dimension in the query vector as we plan to do at the first glimpse, we conserve this dimension extending operation but assign a new random number t to the extended dimension in each query vector. Such a newly added arbitrariness is expected to increase the difficulty for the cloud server to study the association among the received trapdoors. In addition, as mentioned in the keyword privacy requirement ,[7] arbitrariness 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 arbitrariness in the final similarity score is an effective way towards what we expect here. More particularly, unlike the arbitrariness involved in the query vector, we insert a duplicate keyword into each data vector and assign a random value to it. Each individual vector Di is extended to (n+2)-dimension instead of (n + 1), where a arbitrary variable εi representing the duplicate keyword is stored in the extended dimension. The whole approach to achieve ranked search with multiple keywords over encrypted data is as follows:

1. Setup The data owner randomly generates a (n + 2)-bit vector as S and two (n+2)×(n+2) invertible matrices $\{M1,M2\}$. The secret key SK is in the form of a 3-tuple as $\{S,M1,M2\}$.
2. BuildIndex(F, SK) The data owner generates a binary data vector Di for every document Fi, where each binary bit $Di[j]$ represents whether the equivalent keyword Wj occurs in the document Fi. Subsequently, every plaintext subindex $\vec{D}\ i$ is generated by applying dimension extending and splitting procedures on Di. These procedures are similar with those in the protected kNN computation except that the (n + 1)-th entry in $\vec{D}\ i$ is set to a random number εi, and the (n + 2)-th entry in $\vec{D}\ i$ is set to 1 during the dimension extending. $\vec{D}\ I$ is therefore equal to (Di, εi, 1). Finally, the subindex Ii = $\{MT1\ \vec{D}i',MT2\ \vec{D}i''\}$ is built for every encrypted document Ci.
3. Trapdoor(fW) With t keywords of concern in fW as input, one binary vector Q is generated where each bit $Q[j]$ indicates whether $Wj \in$ fW is true or false. Q is first extended to n + 1-dimension which is set to 1, and then scaled by a random number $r \neq 0$, and finally extended to a (n + 2)-dimension vector as $\vec{Q}$ where the last dimension is set to another random number t. $\vec{Q}$ is therefore equal to (rQ, r, t). After applying the same splitting and encrypting processes as above, the trapdoor TfW is generated as $\{M^{-1}\ 1\ \vec{Q}',M^{-1}\ 2\ \vec{Q}''\}$.
4. Query(TfW, k, I) With the trapdoor TfW, the cloud server calculates the comparison scores of each document Fi as in equation 1. WLOG, we assume $r > 0$. After sorting all scores, the cloud server returns the top-k ranked id list FfW. $\vec{}$ Note that in the original case, the final score is simply rDi

V. ANALYSIS

Index Construction: To build a searchable sub index Ii for each document Fi in the dataset F, the first step is to record the keyword set extracted from the document Fi to a data vector Di, followed by encrypting every data vector. The time cost of mapping based on directly on the dimensionality of data vector which is determined by the amount of the dictionary, i.e., the number of indexed keywords. And the time required to build the whole index is also related to the number of sub index which is equal to the number of

documents in the dataset. Fig. 2(a) shows that, specified the same dictionary where $|W| = 4000$, the time required to build the whole index is nearly linear with the size of dataset since the time cost of building each sub index is fixed. Fig. 2(b) shows that the number of keywords indexed in the dictionary calculates the time cost of building a sub index. The major computation to generate a sub index in MRSE I includes the dividing process and two multiplications of a $(n + 2) \times (n + 2)$ matrix and a $(n + 2)$ *2)*

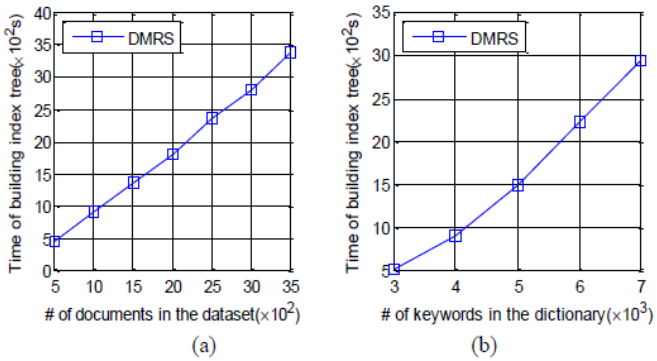

Figure 2(a): Time cost of building Index for n=4000
Figure 2(b): Time cost of building Index for m=1000

Trapdoor Generation: Fig. 3(a) shows that the time to generate a trapdoor is greatly affected by the number of keywords in the dictionary. Like index construction, every trapdoor creation has two multiplications of a matrix and a split query vector, where the greatness of matrix or query vector is different in two proposed schemes and becomes larger with the increasing size of dictionary. Fig. 3(b) demonstrates the trapdoor generation cost in the MRSE II scheme is about 20 percentages larger than that in the MRSE I scheme. Like the sub index generation, the difference of costs to generate trapdoors is majorly caused by the different greatness of vector and matrices in the two MRSE schemes. More importantly, it shows that the number of query keywords has little control on the overhead of trapdoor creation, which is a significant benifts over associated works on multi-keyword searchable encryption.

3) Query: Query execution in the cloud server consists of computing and rank of evaluation scores for all documents in the dataset.

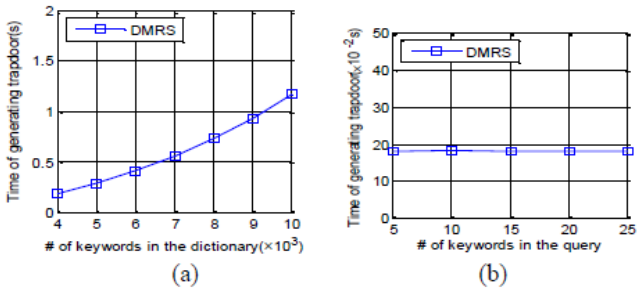

Figure 2(a): Time cost of generating trap door for t= 10
Figure 2(b): for different number of query keywords n= 4000

VI. CONCLUSION

In this paper, a new framework is proposed for the problem of multi-keyword ranked search over encrypted cloud data, and to generate a range of privacy requirements. Among various multi-keyword semantics, the well-organized similarity measure is “coordinate matching”, i.e., as many matches are feasible,to successfully captures the relevance of outsourced documents to the query keywords, and employ “inner product comparison” to evaluate such comparison measure. For meeting the challenge of supporting multi-keyword semantic without security breaches, MRSE framework is proposed using protected inner product calculation. Thorough analysis computing privacy and efficiency guarantees of proposed schemes is given, and experiments on the real-world samples shows our proposed scheme introduces low overhead on both computation and communication.

REFERENCES

[1] K. Ren, C. Wang and Q. Wang, "Security challenges for the public cloud", Internet Computing, IEEE, vol. 16, no. 1, **(2012)**, pp. 69-73.
[2] D. Boneh, E. Kushilevitz, R. Ostrovsky and W. E. Skeith III, "Public key encryption that allows PIR queries", Advances in Cryptology-CRYPTO 2007, Springer, **(2007)**, pp. 50-67.
[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search", Advances in Cryptology-Eurocrypt 2004, **(2004)**, pp. 506-522.
[4] P. Van Liesdonk, S. Sedghi, J. Doumen, P. Hartel and W. Jonker, "Computationally efficient searchable symmetric encryption", Secure Data Management, Springer, **(2010)**, pp. 87-100.
[5] M. Bellare, A. Boldyreva and A. O’Neill, "Deterministic and efficiently searchable encryption", Advances in Cryptology-CRYPTO 2007, Springer, **(2007)**, pp. 535-552.
[6] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data, Security and Privacy, 2000. S&P 2000", Proceedings. 2000 IEEE Symposium on, **(2000)**, pp. 44-55.
[7] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data", Applied Cryptography and Network Security, **(2005)**, pp. 442-455.
[8] R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions", Proceedings of the 13th ACM conference on Computer and communications security, **(2006)**, pp. 79-88.
[9] S. Zerr, D. Olmedilla, W. Nejdl and W. Siberski, "Zerber+ r: Top-k retrieval from a confidential index", Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, **(2009)**, pp. 439-449.
[10] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data", In Theory of cryptography, Springer, **(2007)**, pp. 535-554.