INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue1)
Available online at: www.ijariit.com

A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data

Arati Deshmukh
ME Student, Dept. of Computer Engineering, PK Technical Campus, Pune, India

Dr. S. T. Singh
Dept. of Computer Engineering, PK Technical Campus, Pune, India

Prof. P. B. Sahane
Prof. & H.O.D. Dept. of Computer Engineering, PK Technical Campus, Pune, India

Abstract: ***The major aim of this paper is to solve the problem of multi-keyword ranked search over encrypted cloud data (MRSE) at the time of protecting exact method wise privacy in the cloud computing concept. Data holders are encouraged to outsource their difficult data management systems from local sites to the business public cloud for large flexibility and financial savings. However for protecting data privacy, sensitive data have to be encrypted before outsourcing, which performs traditional data utilization based on plaintext keyword search. As a result, allowing an encrypted cloud data search service is of supreme significance. In view of the large number of data users and documents in the cloud, it is essential to permit several keywords in the search demand and return documents in the order of their appropriate to these keywords. Similar mechanism on searchable encryption makes center on single keyword search or Boolean keyword search, and rarely sort the search results.***

In the middle of various multi-keyword semantics, deciding the well-organized similarity measure of "coordinate matching," it means that as many matches as possible, to capture the appropriate data documents to the search query.

Particularly, we consider "inner product similarity" i.e., the amount of query keywords shows in a document, to quantitatively estimate such match measure that document to the search query. Through the index construction, every document is connected with a binary vector as a sub-index where each bit characterize whether matching keyword is contained in the document. The search query is also illustrates as a binary vector where each bit means whether corresponding keyword appears in this search request, so the matched one could be exactly measured by the inner product of the query vector with the data vector. On the other hand, directly outsourcing the data vector or the query vector will break the index privacy or the search privacy. The vector space model facilitate to offer enough search accuracy, and the DES encryption allow users to occupy in the ranking while the popularity of computing work is done on the server side by process only on cipher text. As a consequence, data leakage can be eradicated and data security is guaranteed.

Keywords: Searchable Encryption, Multi-Keyword Ranked, Search, Dynamic Update, Cloud Computing.

I. INTRODUCTION

Cloud computing is a conversational phrase used to express a variety of dissimilar types of computing ideas that occupy large number of computers that are connected through a real-time communication network i.e. Internet. In science, cloud computing is the capability to run a program on many linked computers at the same time. The fame of the term can be recognized to its use in advertising to sell hosted services in the sense of application service provisioning that run client server software on a remote location. Cloud computing relies on sharing of resources to attain consistency and financial system alike to a utility (like the electricity grid) over a network. The cloud also centers on maximize the effectiveness of the shared resources. Cloud resources are typically not only shared by multiple users but as well as dynamically re-allocated as per demand. This can perform for assigning resources to users in dissimilar time zones. For example, a cloud computing service which serves American users during American business timings with a specific application (e.g. email) while the same resources are getting reallocated and serve Indian users during Indian business timings with another application (e.g. web server).

This mechanism must take full advantage of the use of computing powers thus decreasing environmental damage as well, since less power, air conditioning and so on, is necessary for the same functions. The expression "moving to cloud" also explains to an organization moving away from a traditional CAPEX model i.e. buy the devoted hardware and decrease in value it over a period of time to the OPEX model i.e. use a shared cloud infrastructure and pay as you use it. Proponents maintain that cloud computing Permit Corporation to avoid direct infrastructure costs, and focus on projects that distinguish their businesses as an alternative of infrastructure. Proponents also maintains that cloud computing permit schemes to get their applications should run faster, with better manageability and less maintenance, and enable IT to more quickly adjust resources to meet random and changeable business demand.

II.EXISTING AND PROPOSED SYSTEMS

A. Existing System

A general approach to protect the data confidentiality is to encrypt the data before outsourcing. Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute keyword search over ciphertext domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword Boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi-keyword ranked search achieves more and more attention for its practical applicability. Recently, some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server.

- Huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical.
- Existing System methods not practical due to their high computational overhead for both the cloud sever and user.

B. Proposed System

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (TF) × inverse document frequency (IDF)" model are combined in the index construction and query generation to provide multi-keyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search" algorithm based on this index tree. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known ciphertext model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model.

- Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents.
- We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.
- Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our "Greedy Depth-first Search" algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

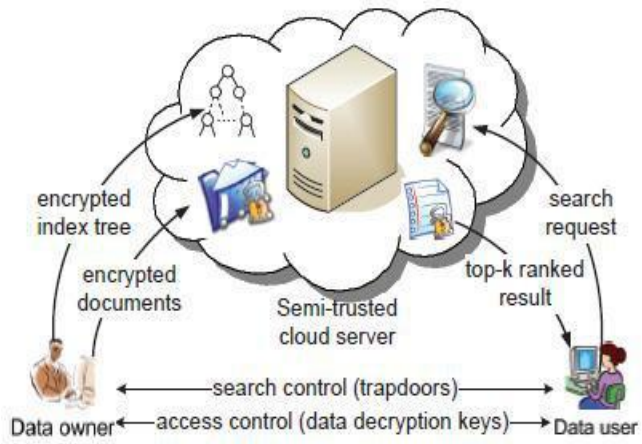

Fig.1. The architecture of ranked search over encrypted cloud data.

III. MULTI-KEYWORD RANKED SEARCH OVER ENCRYPTED (MRSE)

Now a day's cloud computing has become essential for many utilities, where cloud customers can slightly store their data into the cloud so as to benefit from on-demand high-quality request and services from a shared pool of configurable computing resources. Its huge suppleness and financial savings are attracting both persons and enterprise to outsource their local complex data management system into the cloud. To safe guard data privacy and struggle unwanted accesses in the cloud and away from, sensitive data, for example, emails, personal health records, photo albums, videos, land documents, financial transactions, and so on, may have to be encrypted by data holder before outsourcing to the business public cloud; on the other hand, obsoletes the traditional data use service based on plaintext keyword search. The insignificant solution of downloading all the information and decrypting nearby is clearly impossible, due to the enormous amount of bandwidth cost in cloud scale systems. Furthermore, apart from eradicating the local storage management, storing data into the cloud supplies no purpose except they can be simply searched and operated. Thus, discovering privacy preserving and effective search service over encrypted cloud data is one of the supreme importance.

In view of the potentially large number of on-demand data users and vast amount of outsourced data documents in the cloud, this difficulty is mostly demanding as it is really difficult to gather the requirements of performance, system usability, and scalability. On the one hand, to congregate the efficient data retrieval requirement, the huge amount of documents orders the cloud server to achieve result relevance ranking, as an alternative of returning undifferentiated results. Such ranked search system allows data users to discover the most appropriate information quickly, rather than burdensomely sorting during every match in the content group. Ranked search can also gracefully remove redundant network traffic by transferring the most relevant data, which is highly attractive in the "pay-as-you-use" cloud concept. For privacy protection, such ranking operation on the other hand, should not reveal any keyword to related information. To get better the search result exactness as well as to improve the user searching experience, it is also essential for such ranking system to support multiple keywords search, as single keyword search often give up far too common results. As a regular practice specifies by today's web search engines i,e Google search, data users may lean to offer a set of keywords as an alternative of only one as the indicator of their search interest to retrieve the most relevant data. And each keyword in the search demand is able to help narrow down the search result further. "Coordinate matching", as many matches as possible, is an efficient resemblance measure among such multi-keyword semantics to refine the result significance, and has been widely used in the plaintext information retrieval (IR) community. Though, the nature of applying encrypted cloud data search system remains a very demanding task in providing security and maintaining privacy, like the data privacy, the index privacy, the keyword privacy, and many others.

Encryption is a helpful method that treats encrypted data as documents and allows a user to securely search through a single keyword and get back documents of interest. On the other hand, 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 put up such high service-level needs like system usability, user searching experience, and easy information discovery. Even though some modern plans have been proposed to carry Boolean keyword search as an effort to improve the search flexibility, they are still not sufficient to provide users with satisfactory result ranking functionality. The solution for this problem is to secure ranked search over encrypted data but only for queries consisting of a single keyword. The challenging issue here is how to propose an efficient encrypted data search method that supports multi-keyword semantics without privacy violation. In this paper, we describe and solve the problem of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving exact system wise privacy in the cloud computing concept. Along with various multi-keyword semantics, select the efficient resemblance measure of "coordinate matching," it means that as various matches as possible, to confine the significance of data documents to the search query. Particularly, inner product similarity the numbers of query keywords show in a document, to quantitatively calculate such similarity assess of that document to the search query.

For the period of the index construction, each document is associated with a binary vector as a sub-index where each bit signifies whether matching keyword is contained in the document. The search query is also illustrates as a binary vector where each bit means whether corresponding keyword appears in this search request, so the resemblance could be exactly calculated by the inner product of the query vector.

With the data vector. On the other hand, directly outsourcing the data vector or the query vector will go against the index privacy or the search privacy. To face the challenge of cooperating such multi keyword semantic without privacy breaches, we propose a basic idea for the MRSE using secure inner product computation, which is modified from a secure k-nearest neighbor (kNN) method, and then give two considerably improved MRSE method in a step-by-step way to accomplish different severe privacy needs in two risk models with enlarged attack competence.

IV. PERFORMANCE ANALYSIS

We implement the proposed scheme using C++ language in Windows 7 operation system and test its efficiency on a real-world document collection: the Request for Comments (RFC). The test includes 1) the search precision on different privacy level, and 2) the efficiency of index construction, trapdoor generation, search, and update. Most of the experimental results are obtained with an Intel Core(TM) Duo Processor (2.93 GHz), except that the efficiency of search is tested on a server with two Intel(R) Xeon(R) CPU E5-2620 Processors (2.0 GHz), which has 12 processor cores and supports 24 parallel threads. A. Precision and Privacy. The search

precision of scheme is affected by the dummy keywords in EDMRS scheme. Here, the 'precision' is defined as that in: Pk= k′=k, where k′ is the number of real top-k documents in the retrieved k documents .If a smaller standard deviation σ is set for the random.

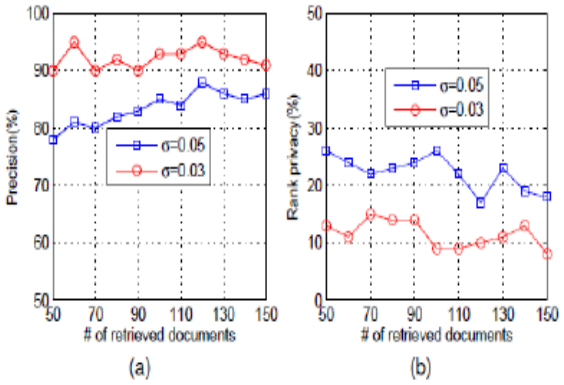

Fig.2. The precision (a) and rank privacy (b) of searches with different standard deviation σ.

Table I. Precision test of basic scheme

NO	Precision	NO	Precision
1	88%	9	96%
2	94%	10	86.7%
3	97%	11	87.5%
4	100%	12	100%
5	85%	13	82.3%
6	89%	14	100%
7	89%	15	100%
8	96%	16	71.1%

Variable Σεv', the EDMRS scheme is supposed to obtain higher precision, and vice versa. The results are shown in Fig.2 (a). In the EDMRS scheme, phantom terms are added to the index vector to obscure the relevance score calculation, so that the cloud server cannot identify keywords by analyzing the TF distributions of special keywords .Here, we quantify the obsceneness of the relevance score by "rank privacy", which is defined as:

$Pk' = \sum |ri - ri'| \; k2$

Where ri is the rank number of document in the retrieved top-k documents, and r′i is its real rank number in the whole ranked results. The larger rank privacy denotes the higher security of the scheme, which is illustrated in Fig. 2(b).

Table II. Storage consumption of index tree

Size of dictionary	1000	2000	3000	4000	5000
BDMRS (MB)	73	146	220	293	367
EDMRS (MB)	95	168	241	315	388

In the proposed scheme, data users can accomplish different requirements on search precision and privacy by adjusting the standard deviation σ, which can be treated as a balance parameter. We compare our schemes with a recent work proposed by Sun et al., which achieves high search efficiency. Note that our BDMRS scheme retrieves the search results through exact calculation of document vector and query vector. Thus, top-k search precision of the BDMRS scheme is 100%. But as a similarity-based multi-keyword ranked search scheme, the basic scheme in suffers from precision loss due to the clustering of sub-vectors during index construction. The precision test of basic scheme is presented in Table I. In each test, 5 keywords are randomly chosen as input, and the precision of returned top 100 results is observed. The test is repeated 16 times, and the average precision is 91%.

B. Efficiency 1. Index Tree Construction:
The process of index tree construction for document collection F includes two main steps:

1) Building an unencrypted KBB tree based on the document collection, and 2) encrypting the index tree with splitting operation and two multiplications of a (m×m) matrix. The index structure is constructed following a post order traversal of the tree based on the document collection, and O(n) nodes are generated during the traversal. For each node, generation of an index vector takes O(m) time, vector splitting process takes O(m) time, and two multiplications of a (m×m) matrix takes O(m2) time. As a whole, the time complexity for index tree construction is O (nm2). Apparently, the time cost for building index tree mainly depends on the cardinality of document collection and the number of keywords in dictionary W. Fig. 3 shows that the time cost of index tree construction is almost linear with the size of document collection, and is proportional to the number of keywords in the dictionary. Due to the dimension extension, the index tree construction of EDMRS scheme is slightly more time-consuming than that of BDMRS scheme. Although the index tree construction consumes relatively much time at the data owner side, it is noteworthy that this is a one-time operation. On the other hand, since the underlying balanced binary tree has space complexity O (n) and every node stores two m-dimensional vectors, the space complexity of the index tree is O (nm). As listed in Table II, when the document collection is fixed (n = 1000), the storage consumption of the index tree is determined by the size of the dictionary.

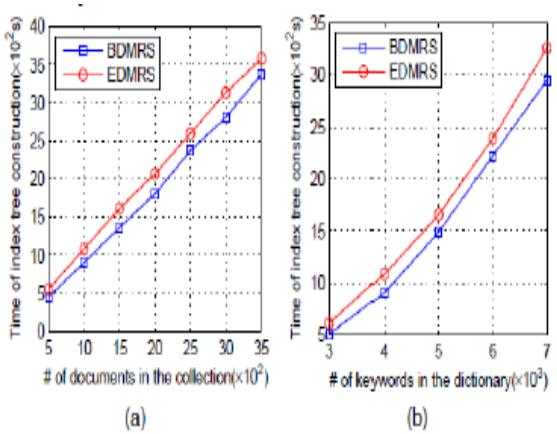

Fig.3. Time cost for index tree construction: (a) for the different sizes of document collection with the fixed dictionary, m = 4000, and (b) for the different sizes of dictionary with the fixed document collection, n = 1000. 2.

Trapdoor Generation: The generation of a trapdoor incurs a vector splitting operation and two multiplications of a (m × m) matrix, thus the time complexity is O (m2), as shown in Fig. 4(a). Typical search requests usually consist of just a few keywords. Fig. 4(b) shows that the number of query keywords has little influence on the overhead of trapdoor generation when the dictionary size is fixed. Due to the dimension extension, the time cost of EDMRS scheme is a little higher than that of BDMRS scheme.

3. Search Efficiency: During the search process, if the relevance scores at nodes is larger than the minimum relevance score in result list R List, the cloud server examines the children of the node; else it returns. Thus, lots of nodes are not accessed during a real search. We denote the number of leaf nodes that contain one or more keywords in the query as θ. Generally, θ is larger than the number of required documents k, but far less than the cardinality of the document collection n. As a balanced binary tree, the height of the index is maintained to be log n, and the complexity of relevance score calculation is O (m). Thus,

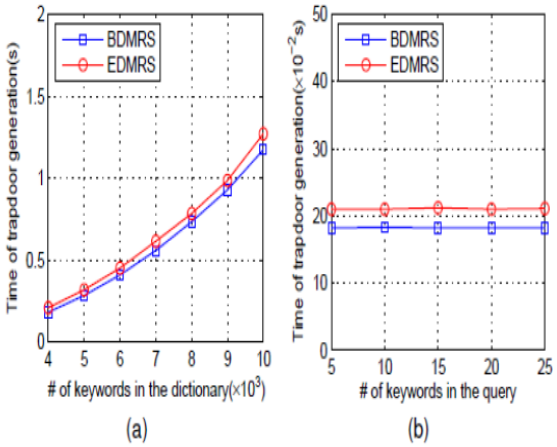

Fig.4. Time cost for trapdoor generation: (a) for different sizes of dictionary with the fixed number of query keywords, t = 10, and (b) for different numbers of query keywords with the fixed dictionary, m = 4000

The time complexity of search is O(θmlog n). Note that the real search time is less than θmlog n. It is because1) many leaf nodes that contain the queried keywords are not visited according to our search algorithm and 2) the accessing paths of some different leaf nodes share the mutual traversed parts. In addition, the parallel execution of search process can increase the efficiency lot. We test the search efficiency of the proposed scheme on a server which supports 24 parallel threads. The search performance is tested respectively by starting 1, 4,8 and 16 threads. We compare the search efficiency of our scheme with that of Sun et al. In the implementation of Sun's code, we divide 4000 keywords into 50 levels. Thus, each level contains 80 keywords. According to, the higher level the query keywords reside, the higher the search efficiency is. In our experiment, we choose ten keywords from the 1st level (the highest level, the optimal case) for search efficiency comparison. Fig. 5 shows that if the query keywords are chosen from the1st level, our scheme obtains almost the same efficiency as when we start 4 threads. Fig5 also shows that the search efficiency of our scheme increases a lot when we increase the number of threads from 1 to 4.

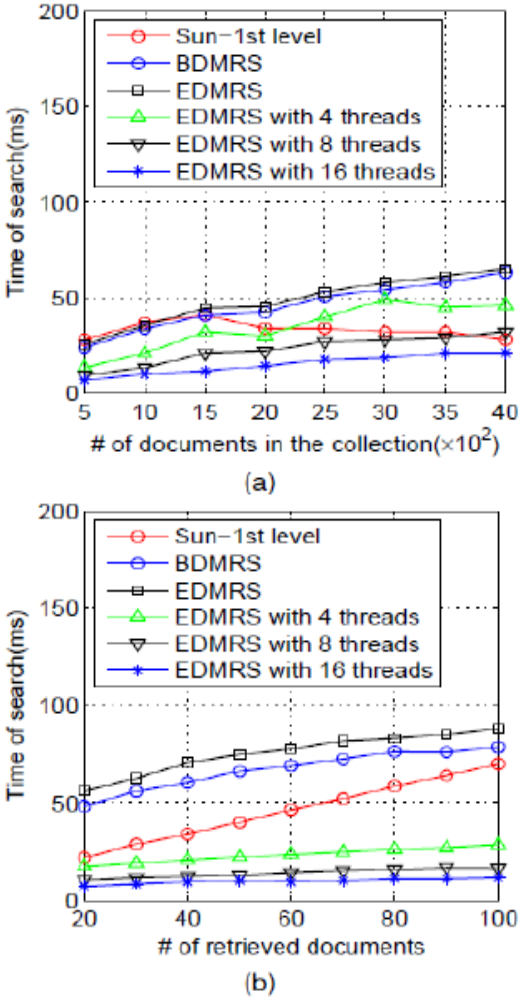

Fig.5. The efficiency of a search with ten keywords of interest as input :(a) for the different sizes of document collection with the same dictionary, m = 4000, and (b) for different numbers of retrieved documents with the same document collection and dictionary, n=1000, and m = 4000**.**

However, when we continue to increase the threads, the search efficiency is not increase remarkably. Our search algorithm can be executed in parallel to improve the search efficiency. But all the started threads will share one result list R List in mutually exclusive manner. When we start too many threads, the threads will spend a lot of time for waiting to read and write the R List. An intuitive method to handle this problem is to construct multiple result lists. However, in our scheme, it will not help to improve the search efficiency a lot. It is because that we need to find k results for each result list and time complexity for retrieving each result list is O(θmlog n=l). In this case, the multiple threads will not save much time, and selecting k results from the multiple result list will further increase the time consumption. In the Fig. 6, we show the time consumption when we start multiple threads with multiple result lists. The experimental results prove that our scheme will obtain better search efficiency when we start multiple threads with only one result list.

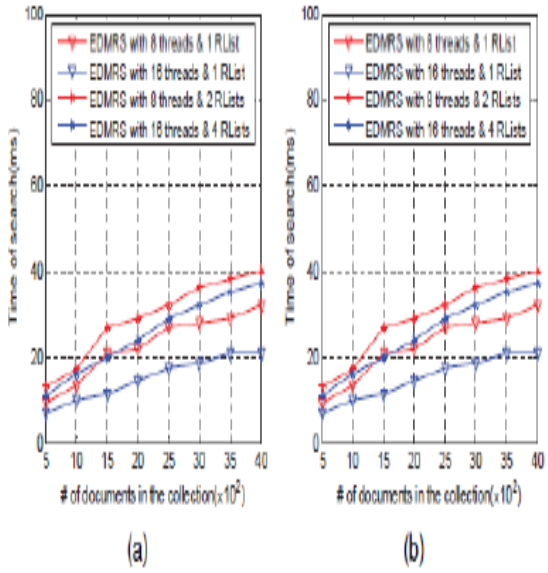

Fig.6. the efficiency of a search with ten keywords of interest as input: (a) for the different sizes of document collection with the same dictionary, m = 4000, and (b) for different numbers of retrieved documents with the same document collection and dictionary, n = 1000, and m = 4000.

4. Update Efficiency: In order to update a leaf node, the data owner needs to update log n nodes. Since it involves an encryption operation for index vector at each node, which takes O(m2) time, the time complexity of update operation is thus O(m2 log n). We illustrate the time cost for the

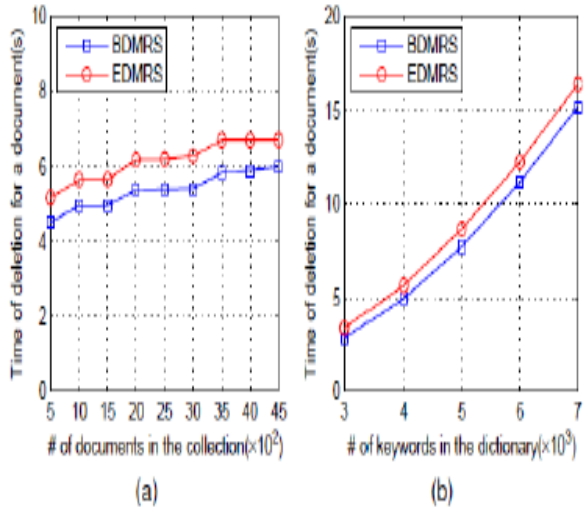

Fig.7. Time cost for deletion of a document: (a) for the different sizes of document collection with the same dictionary, m = 4000, and (b) for the same document collection with different sizes of dictionary, n = 1000.

Deletion of a document. Fig. 7(a) shows that when the size of dictionary is fixed, the deletion of a document takes nearly logarithmic time with the size of document collection. And Fig. 7(b) shows that the update time is proportional to the size of dictionary when the document collection is fixed. In addition, the space complexity of each node is O (m). Thus, space complexity of the communication package of updating a document is O (mlog n).

V. CONCLUSION

In this paper we describe and solve the problem of multi-key word ranked search over encrypted cloud data, and set up a range of privacy requirements. Among various multi-keyword semantics, we select the efficient similarity measure of "coordinate matching," i.e., as many equivalent as possible, to effectively capture the Relevance of outsourced documents to the query Keywords, and utilize "inner product similarity" to quantitatively calculate such comparison measure. In order to acquire the test of supporting multi-keyword semantic without privacy violation, we offer a basic idea of MRSE using secure inner product calculation. Then, we give two improved MRSE schemes to attain various severe privacy needs in two different threat models. The further enhancements of our ranked search method, including supporting more search semantics, i.e., TF×IDF, and dynamic data process detailed analyses in

investigating privacy and efficiency assurance of proposed schemes are mentioned, and testing on the real-world data set demonstrate our proposed schemes which introduces low transparency on both calculation and communication.

REFERENCES

[1] Zhihua Xia, Member, IEEE, Xinhui Wang, Xingming Sun, Senior Member, IEEE,and Qian Wang, Member, IEEE, "A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data", IEEE Transactions on Parallel and Distributed Systems Vol: Pp No: 99 Year 2015.

[2] K. Ren, C.Wang, Q.Wang et al.,"Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[3] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136–149.

[4] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[5] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM (JACM), vol. 43, no. 3,pp. 431–473, 1996.

[6] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology-Eurocrypt 2004. Springer, 2004, pp. 506–522.

[7] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III,"Public key encryption that allows pir queries," in Advances inCryptology-CRYPTO 2007. Springer, 2007, pp. 50–67.

[8] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques forsearches on encrypted data," in Security and Privacy, 2000. S&P2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44–55. [9] E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrint Archive,vol. 2003, p. 216, 2003. [10] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proceedings of the Third international conference on Applied Cryptography and Network Security.Springer-Verlag, 2005, pp. 442–455.

[11] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions, "in Proceedings of the 13th ACM conference on Computerand communications security. ACM, 2006, pp. 79–88.

[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

