ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 2, February 2018

Multi-Keyword Ranked Search over Encrypted Cloud Data Using Bloom Filters and Blind Storage

[1] Sana Shaikh, [2] Khan Rahat Afreen
[1][2] Computer Science and Engineering Department, Deogiri Institute Engineering & Management Studies
Aurangabad, India

Abstract: **Today most organizations prefer to outsource their data on cloud. The outsourced documents and files should be encrypted because of the protection and secrecy worries of their proprietor. As a large amount of data from various clients is getting accumulated on cloud, this raises the issue of security and privacy to its proprietors. Data being large, quick efficient and authorized search is a challenge. An efficient multi-keyword ranked search scheme is proposed in this paper that is able to address the aforementioned problems. Bloom filters are used to enhance search duration. Relevance scoring technique is used to generate ranking results in view of the top-k precision. Inside of this framework, we implemented the blind storage technique to cover access pattern of the search user. Till now the search authorization problem was not considered, that is the cloud server only has to return the search results to authorized users. In this paper, we propose an authorized and ranked multi-keyword search scheme over encrypted cloud data. Identity Based-authentication is used for authentication with AES for encryption. As a result, information leakage can be eliminated and data security is ensured. Security and performance analysis show that the proposed scheme can achieve much improved efficiency in terms of accuracy, search time and security compared with the search algorithm used in EMRS i.e Efficient Multi-keyword Rank Search scheme.**

Index Term: **Cloud Computing, Bloom Filter, Relevance Scoring, Search Authorization, Blind storage, Identity Based authentication, AES**

I. INTRODUCTION

Cloud computing can be described as a single computing machine or more number of machines which as a server/s, holds all necessary applications and report(information) all at one place so that any user can access it from anywhere whoever has an access to that server without actually using his/her physical machine [1]. There are many cloud platforms present in the market like Google Drive, cloud; SkyDrive, Amazon S3, Dropbox and Microsoft Azure which provide storage services. Security and privacy concerns are the main issues in cloud computing as computing resources is shared by many users. To meet practical search requirements, search over encrypted data should have these three following functionality, first is providing 'Data user' the same searching experience over encrypted data as he can have over plaintext data. Second the cloud server should sort the search result in ranked relevance order of the document requested by 'Data user' also eliminating the redundant network traffic by only sending back the most relevant top-k results. Third, for search efficiency, as number of documents outsourced to the cloud could be large, there should be quick respond to the search requests with minimum delays.

[B] PROPOSED METHODOLOGY

The proposed system developed for the searchable encryption for multi-keyword ranked search utilize the Relevance score ,Identity-Based Authentication, Bloom filters, AES and Blind Storage to achieve efficient Multi-Keyword Ranked Search using Bloom filters and Blind storage (MKRSBB) that can return the ranked search results based on the accuracy. Relevance Scoring: Due to a large number of Files stored on cloud, search results should be retrieved in an order of the relevancy with the searched keywords. Scoring is the expected way to weight the relevancy of the documents .Amongst many scoring techniques; we adopt TF-IDF weighting [2] in the MRSBB. In TF-IDF weighting, term frequency $[\![tf]\!]\_(t,f)$ refers to the number of term t in a document f. Inverse document frequency $[\![idf]\!]\_t$ denotes the number of documents which contain term t. Then, the weighting of term t in a document f can be calculated as $[\![tf]\!]\_(t,f) * [\![idf]\!]\_t$ Bloom Filters: Bloom Filters [3] is a kind of probabilistic data structure which is highly space efficient and can be used for instant search. We implement "bloom filters" which uses m-bit array for representation of collection and determine whether an element is member of collection or not, which means it may sometimes gives false positives (error), but promises returns no false negatives. Using counting bloom filters and relevance scoring thus provides efficient multi-keyword ranked search. Initially it is set to 0 in all positions for a given set $S = \{ a\_1 , a\_2 \ldots a\_n\}$, use k independent hash functions from $H = \{ h\_i \mid h\_j : S \rightarrow m, j \leq i \leq k\}$ with each entry of an element $a \in S$ into the Bloom filter a counter is incremented. To check whether a keyword j is in S, feed it to each of the l hash functions to get l array positions. If the bit at any position is 0, $j \notin S$; otherwise, either $j \in S$ or j yields a false positive.

Blind Storage: The proposed scheme also builds a blind storage system [4] on the cloud server to support adding, updating and deleting documents and concealing the pattern accessed by the search user from the cloud server is done by dividing all documents into fixed sized blocks or chunks. The files related seed are used to generate random number which is used to index the chunks. Search Authorization: The documents that are outsourced by the Data owner are sensitive and confidential; these documents should be only provided to the authorized users. In some cases, some unauthorized users may also feel interest to access the document that are outsourced by the Data owner. If they succeed to access these documents then the integrity of data may be violated. So it is very important to save the documents from the access of unauthorized user. In other words, all users do not have the right to access all the documents in the cloud. Every user has their own privileges to access the document from the cloud as per the data owner's authority. Identity Based Authentication: The proposed scheme uses the Data User's email id as unique personal identity for generating the token. Other authentication scheme for Multi-keyword ranked search presented in [5] [6] and [8] is based on the association with registration authority. Identity Based Authentication (IBA) systems are variants which attempt to eliminate one major problem of the traditional approaches: The need to only obtain the receiver's id and password. IB authentication are based on the idea that the token of an entity can be generated from users personal id, unique identifier, e.g. an e-mail address etc which is verified by sending token on it. The Data user will be able to search and download the authorized file only when the combination of Data user's public id i.e e-mail id, password and token is provided.

[C] Experimental Setup:

1. Data User Module:

Data users can download files from the cloud that are uploaded by the data owners. Since the files stored on the cloud server can be in large numbers, thus user should be able to do a multi-keyword search on the cloud server. The authorized user will get results for the specific search, to whom data owners will provide privilege of accessing that file.

2. Data Owner Module:

The data owners can upload the documents to cloud. The documents are encrypted before being outsourced to the cloud. The data owners are provided the option to search the document using keywords of the file that are uploaded to the server. These keywords are used for the indexing purpose due to which the search return values very quickly. These files when once available on the cloud, the data users can search using the keywords. The data owners will also be provided with a file authority option so that Data owner can specify access of files to authorized user with whom he wants to share the file.

3. File Upload & Encryption Module:

This module will encrypted the file before they are outsourced to the cloud. These encrypted files when once outsourced on the cloud, they are split in to chunks so that the cloud server does not recognize access pattern of user. All documents are divided into fixed-size chunks. The encryption of these documents uses AES algorithm so that unauthorized users will not be able to download these files. This ensures the files to be protected from unauthorized user.

4. File Download & Decryption Module:

The data users will be provided by set of keywords to inform them that data owner had outsourced a file for them and that particular data user is authorized to search and download the file. Set of keywords will notify the data owner has authorized him with specific files. If the data owner has enabled his authority, the users should be able to search, download the decrypted file. Before downloading the file, data user will have to be decrypted with a key. This key will be provided to the data users on his account through the trap-door request. For decryption when the key is provided then data users will get the access of the file. The data users can search the files with multi-keyword rank searching methodoly.

[D] PERFORMANCE EVALUATION:

We have evaluated the performance of proposed system on Intel CORE i5 CPU with processing speed of 1.7Ghz and 4GB of RAM running on Windows 10 operating system. The experiment result shows Bloom Filter generated time cost for Index construction and trapdoor generation and increase in result accuracy, search efficiency, security and functionality. We apply a real dataset National Science Foundation Research Awards Abstracts 1990-2003 [10] by selecting and uploading large numbers of documents from it.

1. Index and Trapdoor Generation:

1.1 Index construction:

Index construction in the MKRSBB consists of two phases:

1) Building Index based on the keywords of file collection F provided by data owner

2) Encrypting the index with splitting operation and providing it to Bloom Filter.

The final secure index for each file is a Bloom filter that contains all of the keywords in the file in which the keywords

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 2, February 2018

are first transformed into its hash vector representation and subsequently inserted into the Bloom filter by MD5 functions. Because the query and the index are constructed in the same way, we can compute the relevance of the query to each file. If a document contains the keyword(s) in the query, the corresponding bits in both vectors are 1, thus the inner product is a high value. Finally, the top-K files will be returned. Because Bloom filter generation was linear in the number of the keywords, the index vector generation time could be large, but it was just a one-time effort.

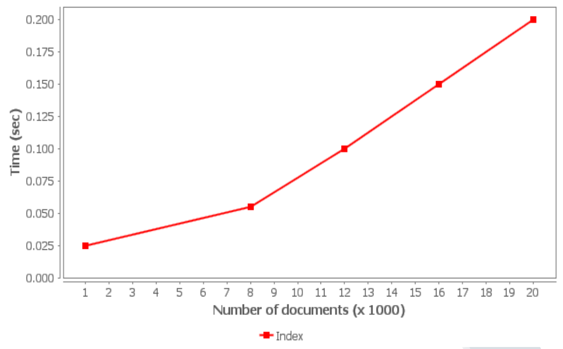


Figure 1.1 Time for Index Construction for different number of documents and number of searched keyword |Q|= 20

Figure 1.1 shows the index generation time for keywords of files for different size of documents. The index structure we constructed is a per document based index. As for the computation of encrypted relevance vector, the data owner first needs to compute the relevance score for each keyword in each document using the TF − IDF technique. Then, to compute the encrypted relevance vector P, the data owner needs to compute index and query vector and a (d+1)-dimension vector with complexity $O((d+1)^3)$. The time cost for computing all the encrypted index vectors is linear to the size of the dataset since time for building sub index of one document is fixed. So, the computation complexity of encrypting a index vector is $O(m(d+1)^3)$ where m represents the number of documents in the dataset and d represents the size of the keyword dictionary |W|.

1.2 Trapdoor generation:

In MKRSBB, trapdoor consists of stag and query generated by the data user which is encrypted and passed to the Bloom filter. Bloom Filter then add it to comparison function generation.

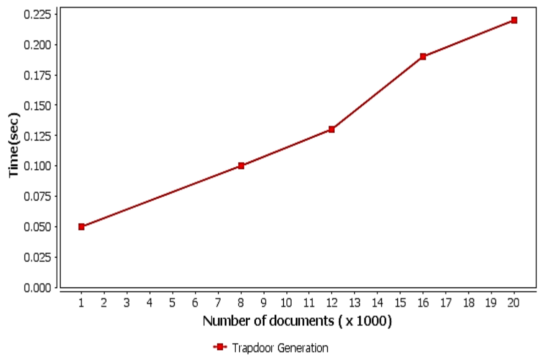


Figure 1.2 Time for Trapdoor Generation for different number of documents and number of searched keyword |Q|= 20

Figure 1.2 shows the total time of trapdoor generation. The trapdoor generation time increased linearly with respect to the number of the inserted query keywords. As the number of keywords grew, the query generation time also increased. As for computing the encrypted query vector |Q|, the Bloom Filter needs to compute comparison of index and query vector and a (d+2)-dimension vector with complexity O(d). The computation complexity of trapdoor generation for the search user is O(d). Thus, the computation complexity is O(md), where m represents the number documents in the dataset and d represents the size of the keyword dictionary |W|.

II. SEARCH EFFICIENCY

The search time in MKRSBB scheme linearly increases with the scale of the dataset, which is impractical for large-scale dataset. Upon receiving stag and encrypted query, Bloom Filter have a instant comparison on the hash values of query encrypted query and index providing quick results and the cloud server can then use stag to access blind storage and retrieve the encrypted blocks. These blocks consist of blocks of documents containing the stag-related keyword. Thus, the MKRSBB can significantly decrease the number of documents which are relevant to the searched keywords.

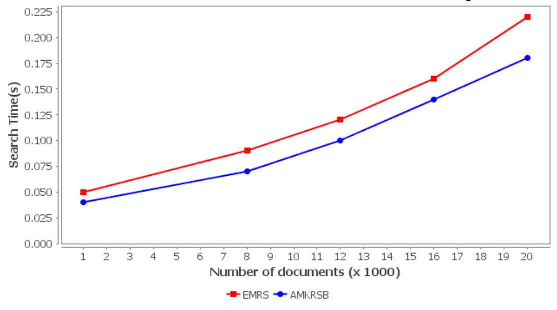


Figure 1.4 The search time for different number of the document set and number of searched keyword |Q|= 20

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 2, February 2018

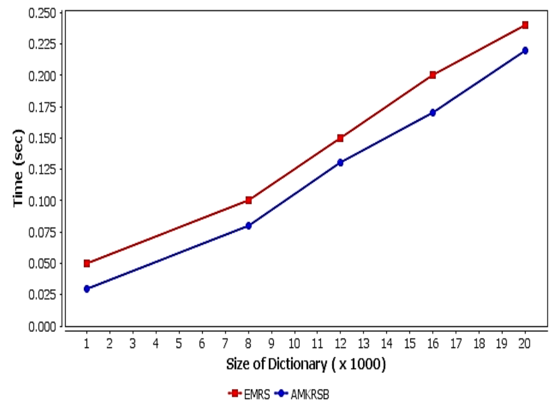


Figure 1.5 The search time of different size of dictionary and number of searched keyword, m=8000,|Q|=20.

One important parameter that affected the search time is the number of the files n because our index was a per file based index, the search time increased linearly in the number of files, as illustrated. As shown in Figure 1.4, the computation cost of search phase is mainly affected by the number of documents in the dataset and the size of the keyword dictionary. So the search time is mainly affected by two factors: the size of index for dataset, and the size of query vector. The search time complexity is O(kd), as the cloud server need to check each document for a search request where k represents comparison in Bloom Filter of hash values of the index vector I for document satisfies the search query vector |Q| ; and d represents computing the relevance score between I and |Q|.

For each search request |Q|, the search procedure implements as follows:

So the search time is mainly affected by two factors: the size of index for dataset, and the size of query vector. The search time complexity is O(kd), as the cloud server need to check each document for a search request where k represents comparison in Bloom Filter of hash values of the index vector I for document satisfies the search query vector |Q|; and d represents computing the relevance score between I and |Q|. The computation complexity for search operation in the EMRS is O(αzd), where z represents the number of documents which contain the keyword applied by the keyword-related token stag and the α is the extension parameter that scales the number of blocks in a document to the number of blocks in the set Sf .

The computation cost of search phase is mainly affected by the number of documents in the dataset and the size of the keyword dictionary. Note that, although the time costs of search operation are linearly increasing in both schemes, the increase rate of the MKRSBB is less than that of EMRS scheme.

IV. SECURITY ANALYSIS:

We analyze the MKRSBB scheme according to the five predefined privacy requirements in the design goals:

Authentication: Reliable authentication mechanism is provided in MKRSBB so that any unauthenticated user should not try to access the sensitive data of the Data owner which he wants to share with the Data User.

Search Authorization: The search user with attributes that satisfy the access policy defined by data owners can search and download the authorized cipher text to get the content key and further decrypt the encrypted document. Thus, the secrecy of documents is well protected in the MKRSBB.

Files and Index Confidentiality: In the proposed MKRSBB scheme, encryption is performed on both Documents and index before they are being uploaded to a cloud server. Thus, the MKRSBB scheme is resilient against cipher text-only attack and the index confidentiality and the query confidentiality are well protected.

Trapdoor Privacy: The data user will want that her search should not be exposed to the cloud server. Thus cloud server should be not permitted from knowing the keywords enclosed in the trapdoor of the data user.

Trapdoor Un-linkability: The trapdoors must not be linkable; even when it contains the exact keywords the trapdoors should be completely different. In other words, rather than being randomized it should be deterministic and should not deduce any links between two trapdoors.

Concealing Access Pattern of the Data User: In the MKRSBB, the access pattern should be completely covered from the cloud server. In particular, the cloud server should not get to know the total number of the files stored on it nor the size of searched document even if the data user retrieves this file from the cloud server. The comparison of security level is shown in Table 4.2. We can see that the MKRSBB can achieve best security guarantees compared with the exiting schemes [11], [12],[13]

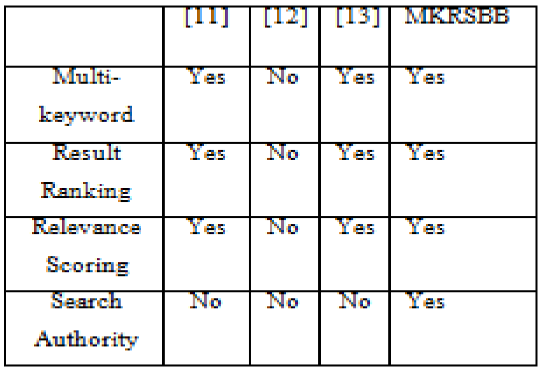
Table 4.2: Comparison of security level

	[11]	[12]	[13]	MKRSBB
IB-Authentication	No	No	No	Yes
Confidentiality of Documents	Yes	Yes	Yes	Yes
Trapdoor Unlink ability	Yes	No	Yes	Yes
Concealing Access pattern of user	No	Yes	Yes	Yes

V. FUNCTIONALITY ANALYSIS:

As there are huge number of documents and data users in a cloud environment, MKRSBB should allow privacy-preserving multi-keyword search and return documents in a order of higher relevance to the search request.

Table 4.2: Comparison of functionalities.



	[11]	[12]	[13]	MKRSBB
Multi-keyword	Yes	No	Yes	Yes
Result Ranking	Yes	No	Yes	Yes
Relevance Scoring	Yes	No	Yes	Yes
Search Authority	No	No	No	Yes

Cao's scheme can have multi-keyword search and returns files in a relevance-based order [11]. Naveed's scheme uses the blind storage system to prevent the access pattern but data user can only have a single-keyword search and returns undifferentiated results [12]. EMRS scheme achieves multi-keyword search, and relevance sorting by preserving a high security but it does not provide search authority to user[13]. MKRSBB can achieve multi-keyword ranked instant search using Bloom Filter, relevance scoring with high security encryption and search authority.

VI. CONCLUSION

We outline and solve the matter of multi-keyword ranked search over encrypted cloud data by using Bloom filters, Blind storage technique and generated authorized result ranking through relevance score technique with minimum delay. The experimental result proves that the proposed architecture not only properly solves the multi-keyword ranked search problem, but also brings an improvement in search accuracy and efficiency, functionality and security level.

REFERENCES

[1] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, ''Secure ranked keyword search over encrypted cloud data,'' in Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst. (ICDCS). 2010 .pp.253-262

[2] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, ''Toward secure multi keyword top-k retrieval over encrypted cloud data, ''IEEE Trans. Dependable Secure Computer., vol. 10, no. 4, pp. 239–250, Jul./Aug. 2013.

[3] Zhihua Xia, Li Chen, Xingming Sun, and Jin Wang "An Efficient and Privacy-Preserving Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data" International Journal of Security and Its Applications Vol.8, No.2 (2014), pp.323-332

[4] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, ``Secure kNN computation on encrypted databases," in Proc. ACM SIGMOD Int. Conf. Manage Data, 2009, pp. 139-152.
[5] W. Sun, et al., ``Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. 8th ACM SIGSAC Symp. Information, Computer. Communication .Security., 2013, pp. 71-82.

[6] Vanishree R , Mr.G.S Suresh "Multi Keyword Ranked Search over Encrypted Cloud Data" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 5, May 2015 pp.2245-2248

[7]D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, ''Highly-scalable searchable symmetric encryption with support for Boolean queries,'' in Proc. CRYPTO, 2013, pp. 353–373.

[8] H. Pang, J. Shen, and R. Krishnan, ``Privacy-preserving similarity-based text retrieval," ACM Trans. Internet Technol., vol. 10, no. 1, p. 4, 2010.

[9] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, ``An SMDP-based service model for inter domain resource allocation in mobile cloud networks," IEEE Trans. Veh. Technol., vol. 61, no. 5, pp. 2222-2232, Jun. 2012.

[10] NSF Research Awards Abstracts 1990-2003. UCI KDD Archive.
[11] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, ''Privacy-preserving multi keyword ranked search over encrypted cloud data,'' IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 222–233, Jan. 2014.

[12]Muhammad Naveed; Manoj Prabhakaran; Carl A. Gunter "Dynamic Searchable Encryption via Blind Storage" 2014 IEEE Symposium on Security and Privacy pp.639-654

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 2, February 2018

[13] Hongwei li, Dongxiao liu "Enabling Efficient Multi-keyword Ranked Search over Encrypted Mobile Cloud Data" IEEE Transactions On Emerging Topics In Computing, 2015,pp 127-137

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