ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

A Dynamic and Efficient Multi-Keyword Ranked Search over Encrypted Cloud Data Using Secure Searching Encryption Algorithm

Sindhu Janjyam, T. Ashok

P.G Student, Dept. of CST., Adikavi Nannya University, Rajamahendhravarm, India

Assistant Professor, Dept. of CST., Adikavi Nannaya University, Rajamahendhravaram, India

ABSTRACT: By the advantages of the cloud computing software as a service, many companies and large scale organization are motivated to outsource their valuable data to the cloud databases for its costless service, easily scalable, and able to access anytime from anywhere, minimizes the cost of management. Due to privacy issues of data in public cloud databases, data should be encrypted before outsourcing to cloud environment. However, searching on outsourced data is obsolete. Data utilization requires keyword based document retrieval. In this, paper we proposed a scheme to support Multi-keyword Ranked Search and simultaneously allows the data owner to perform dynamic operations like insertion and deletion of documents. For that, we use the TF&IDF model in index construction and query generation. Searchable encryption is the cryptographic method used for providing security, for that we explore some effective cryptographic algorithm is CRSA and construct a special tree called balanced M-Way search tree to propose "Greedy-Depth First Search" process to provide Multi-keyword Ranked Search, meanwhile calculating the relevance score between encrypted index and query vectors to providing Top-k Ranked results.

KEYWORDSs: Cloud computing, Searchable encryption, Multi-keyword ranked search, Dynamic operations, Depth-first search technique (DFST), Commutative RSA (CRSA).

I. INTRODUCTION

Cloud computing is the embryonic technology; it is a new model of enterprise IT infrastructure. Cloud computing providing several resources (databases, cpu cycles, network bandwidth, applications, software's) as services over internet. It provide convenient and on demand service to both individuals and organizations for reduces the economic overhead, its scalability and less hands on management. By these appealing features, more and more organizations and individuals are motivated to outsource their sensitive information such as e-mails, health records, and company financial data, documents related to new product releases and government documents, certificates into cloud servers.

There are many cloud service providers in the market to providing the on demand service in pay-as you-use model with less cost, for example Amazon web services, Google app engine, Microsoft azure, Sales force, IBM cloud services are substantially adding to their databases. These services are shared among many users and CSP's (Cloud Service Providers). They keep sensitive and user's outsourced data for user availability with effective recovery mechanisms. To providing confidentiality for user's documents, data should be encrypted with different security keys before outsourcing cloud environment, because we don't say all the data users are honest or a user unfotunatly may lead to modify the content of other user's document. To check user authority we preferably use the authentication mechanisms like password based user logins.

To address these problems, researchers are proposed encryption methods to contributing searching process in cloud. Those searching algorithms are referred as searchable encryption (SE) schemes; they have made specific contributions in terms of efficiency, functionality, security and searching. It should not leak any keyword information to the cloud server. It is very suitable in "pay-as-use" model. In this paper, proposed methods are intended to solve

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

some problems of multi-keyword ranked search over encrypted cloud data and security by preserving strict system wise privacy in cloud environment.

The remainder of this paper is organized as Related work/literature survey is discussed in section 2, section 3 describes the problem formulation with system models and section 4 contains proposed search, encryption schemes and section 5 shows the results, section 6 gives conclusion and future work, references are discussed in later section and figure 1 shows the architecture and figure 2,3 shows the results.

II. LITERATURE SURVEY

The data encryption is used to protect the data confidentiality. Searchable encryption schemes enable the data owner's to store encrypted in cloud server and execute keyword search over cipher text domain. When come to searching on cipher text, the efficiency gets low. This inefficiency made lead to leakage of information to unauthorized people. So, previously many research works have proposed to words for efficient searching over encrypted cloud data.

For the first time, *Song et al.* [1] proposed the first symmetric searchable encryption (SSE) scheme, in this file is encrypted word by word. The drawback of this scheme is word frequency is relieved and search time is linear to size of documents collection. *Change et al.*[2] proposed scheme index is build for each document. This scheme is more secure than Goh's scheme but less efficient and does not support arbitrary update of new words. *Curtmola et al.*[3] for the first time proposed two searchable encryption schemes to achieve the optimal search time. *Kamara et al.* proposed extension to SSE schemes called dynamic SSE, where insertion and deletion of documents can perform in index table, but all these supports only single keyword search. Later there are many searching methods implemented in cloud, such as single keyword Boolean search, which are very simple in functionality, but not efficient from different threat models. After abundant works have been proposed against different threat models to achieve various search functionality, such as single keyword search, exact or fuzzy keyword search, similarity search [1], [2], [3], [4], [5], [6], [7], ranked search [8], [9], [10].

All these proposed multi-keyword ranked search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality. Ranked search can returning most relevant document. Some early works [8],[9],[10], realized that ranked search using order-preserving techniques, but they support single keyword search. *N.Cao et al.*[11] scheme, in which documents and queries are represented as vector of dictionary size with "coordinate matching ", the documents are ranked according to the number of matched query keywords, but this scheme is not accurate enough and search efficiency is O(n). *Sun et al.*[12] scheme supports similarity-based ranking. In this, constructed a searchable index tree based on vector space model and adapted cosine measure together with TF×IDF to provide ranking. This algorithm achieves better than linear search time but results in precision loss. C.O*rencik et al.*[13] proposed a search method, which utilizes the local sensitive hash(LSH) functions to cluster the similar documents. This scheme suitable for similarity search but results cannot give exact ranking. In [14] *Zhang et al.* proposed a method to deal with secure multi-keyword ranked search scheme, in this different data owners use different secret keys to encrypt their documents.

Although researchers from many universities had been investigating to identify a suitable privacy preserving search techniques for cloud domain. There exist wide range of research challenges in searching and privacy for encrypted data, however, we intended to choose the work to meeting their challenges.

III. PROBLEM FORMULATION

A. *Notations and Preliminaries:*

- D: This is collection of plaintext documents stored in cloud server $D=\{d_1,d_2,\ldots\ldots,d_n\}$. Each document is considered as sequence of keywords.
- n:This is the total number of documents in D.
- T: It is the unencrypted form of index tree for the whole document collection D.
- I: Is the searchable encrypted tree index generated from T.
- Q: It is the query vector for keyword set W_q.
- TD: This is referred as trapdoor for search request, it is the encrypted form of Q.
- W: This is the total number of keywords in W.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

- W: It is the dictionary, namely the set keywords, denoted as $W=\{w_1,w_2,\ldots\ldots,w_m\}$.
- M: This is total number of keywords in W.
- W_q: This is the subset of W, representing keywords in the query.
- P_u: Is the index vector stored in tree node u,whose dimensions is equal to cardinality of the dictionary W.
- *Note*: 'u' can be either a leaf node or internal node in the tree.
- C_u: This is encrypted form of P_u.

B. The Vector Space Model for Relevance Score:

The collection of documents can be represented as vectors in vector space model with TF×IDF model supports ranked multi-keyword search efficiently. Here, $TF_{(w,d)}$ is the term frequency defined as the number of times keyword (term) 'w' appears in the document 'd'. the number of documents that contains keyword w is known as document frequency(DF). The inverted document frequency (IDF_w) of a keyword 'w' obtained from the total number of documents is divided by document frequency. Every document is denoted with a vector, whose elements are the normalized TF values of keywords. The query is also denoted with a vector Q, whose elements are normalized IDF values of query keywords. Naturally, the size of query vector and document vector are equal to the total number of keywords in the dictionary. $Q\times I_u$ required to calculate the relevance score between the query and corresponding documents.

- $N_{d,wi}$: The number of keywords w_i in document D.
- N: The total number of documents.
- N_{wi}: The number of documents that contains keyword w_i.
- $TF^1_{d,wi}$: The TF value of w_i in document d.
- IDF_{wi}: The IDF value of w_i in document collection.
- $TF_{u,wi}$: The normalized TF value of the keyword w_i stored in the index vector P_u.
- IDF_{wi}: The normalized IDF value of keyword w_i in document collection.

Relevance score of query vector Q and document vector or index vector P_u of node u is calculated as $Rscore(P_u,Q) = P_u \cdot Q$

$$= \sum_{wi\epsilon W} TFu,wi \times IDFwi \quad (1)$$

If u is an internal node of the tree, $TF_{u,wi}$ is calculated from index vectors in the child nodes of u, if u is a leaf node, the $TF_{u,wi}$ is calculated as:

$$TF_{u,wi} = TF^1_{d,wi} \div \sum wi\epsilon W (TF^1_{d,wi})^2 \text{, Where } TF^1_{d,wi} = 1+\ln N_{d,wi} \quad (2)$$

In the search query vector Q, IDF_{wi} can be calculated as:

$$IDF_{wi} = IDF_{wi} \div \sqrt{\sum wi\epsilon Wq} (IDF^1_{wi})^2 \text{, Where } IDF_{wi} = \ln(1+N/N_{wi}) \quad (3)$$

C. Keyword –based Binary Search Tree:

In this section, we first construct BMs tree based on the vector space model, it is widely used to deal with optimization problems and is a dynamic data structure whose node stores the vector P, and these values are normalized TF values. Later, in algorithm 2 we discuss DFST (depth first search technique) searching algorithm for BMS tree.

D. BMS Tree Index Construction:

In the process of index tree construction, we generate node for each document. These nodes are act as leaf nodes in the tree. Then, the internal nodes are formed based on these leaf nodes. The data structure of the node is defined as u=(ID, D, child[],DID) (4)

Where ID is unique id generated using GenID() function, D I index vector, child[] is pointers to children of the node and DID is a document ID.

In the algorithm 1, $children.D[i] = \max\{u.P_l \rightarrow D[i[],u.P_r \rightarrow D[i]\}$ Where i=1,…., m; (5)

a) Algorithm 1: BuildBMSIndexTree (D)

- *Input:* The document collection D={d1,d2,….,dn} with the identifiers DID={DID\DID=1,2,,…..,n}.
- *Output*: The index tree T.
- for each document d_{DID} in D do
- Construct a leaf node u for d_{DID}, with u.ID=GENID(),$u.P_{l=}u.P_r$=null, u.DID =DID, and $P[i]=TF_{dDI,wi}$, for i=1,2,…..,m;
- Insert u to CurrentNodeSet;

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

- end for
- while the number of nodes in CurrentNodeSet is larger than 1 do
- if the number nodes in CurrentNodeSet is even, i.e 2h then
- for each pair of nodes u' and u" in CurrentNodeSet do
- Generate a parent node u for u' and u", with u.ID=GenID( ),u.P_l=u',u.P_r=u", u.DID=0 and P[i]=max{u'.P[i],u".P[i]} for each i={1,2,…,m}
- Insert u to TempNodeSet;
- end for
- else
- for each pair of nodes u' and u" of the former (2h-2) nodes in CurrentNodeSet do
- Generate a parent node u for u' and u";
- Insert u to TempNodeSet;
- end for
- Create a parent node u_1 for the $(2h-1)^{-th}$ and $2h^{-th}$ node, and then create a parent node u for u_1 and the $(2h+1)^{-th}$ node;
- Insert u to TempNodeSet;
- end if
- Replace CurrentNodeSet with TempNodeSet and then clear TempNodeSet;
- end while

E. Search Process using DFST:

The search process of multi-keyword ranked search scheme is the recursive function upon the BMs tree name as Depth First Search Technique (DFST). We create a result documents as “Ranked list”, whose element is denoted as (Score, DID). The elements of Ranked list are in descending order according to score function during the search process. The DFST algorithm is discussed in algorithm 2.

a) Algorithm 2: DFST (IndexTreeNode u)

- if the node u is not a leaf node then
- if RScore (P_u,Q)>k^{th} score then
- DFST(u.hchild);
- DFST(u.lchild);
- else
- return
- end if
- else
- if RScore (P_u,Q)>k^{th} score then
- Delete the element with the smallest relevance score from Ranked list;
- Insert a new element RScore((P_u,Q), u.DID) and sort all the elements of Ranked list;
- end if
- return
- end if

F. System Models:

In this paper, we explore cloud computing system model involves three different entities. The responsibility of each entity is discussed as follows:

1) Data Owner(DO):

Data owner has a collection of documents D={d_1,d_2,d_3,…..,d_n} with valuable information to be outsourced to the cloud server. To provide data security, the documents are encrypted before outsourcing. Data owner creates the dictionary based on keywords extracted from each document and creates the query vector by using the keywords entered by the data user is called trapdoor. To provide user privacy, query is encrypted and sends it to the data user. The data owner sends access control keys to the authorized user.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

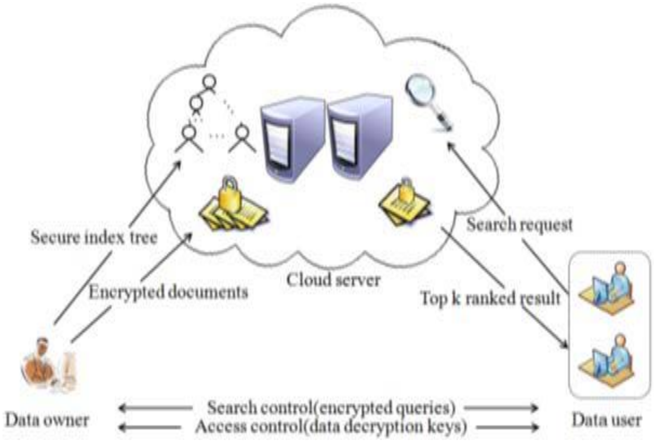

Figure 1: Architecture of multi-keyword ranked search over encrypted cloud data.

2) *Data Users:*

Data users are the authorized users who accessing cloud services and sensitive data from the cloud. The data user create the request for certain documents and send it to data owner, then data user receives trapdoor according to search query request and access control from the data owner. Data user sends these trapdoor and access controls to the cloud server to retrieve documents from the cloud.

3) *Cloud Service Provider:*

Cloud server receives the encrypted documents and encrypted index vectors from data owner and stores into the data owners cloud database. Cloud sever having the capability to take the data request from users and checks the access controls and searches keywords related documents. It will retrieve the documents from cloud storage depending upon the privileges. To increasing document retrieval accuracy from cloud server, the top scored (ranked) documents return to data user. Fig.1 shows the architecture of multi-keyword ranked query search over encrypted cloud data.

G. *Threat Models:*

The cloud server is measured as "honest-but-curious" in our proposed scheme. We considered two threat models for identify the system security and performance in different attack capabilities that is as follows:

a) *Known cipher text model*: In this model, the cloud server knows only about encrypted documents and encrypted index vectors, which are outsourced from data owner.
b) *Known background model*: As compared with known cipher text model, the cloud server is much stronger because it knows about term frequency (TF) statistics of specific keywords in the document collection.

1) *Design Goals:* To enable system secure, our proposed system has following design goals.

a) *Dynamic:* The proposed scheme is designed to not only supporting multi-keyword query search and accurate result ranking, but it also supporting dynamic update on document collection.
b) *Search efficiency:* The scheme aims to achieve sub-linear search efficiency by exploring a special tree based index and an efficient search technique.
c) *Privacy-preserving search:* The scheme is designed to prevent the loud server from learning additional information about the document collection, index tree and query.

The specific privacy requirements are summarized as follows:

a) *Index confidentiality and query confidentiality:* the underlying plaintext information including keywords in the index and query, TF values of keywords in index and IDF values of query keywords should be protected from cloud server.
b) *Trapdoors unlinkability*: the cloud server should not be able to determine whether two encrypted queries (trapdoor) are generated from the same search request.
c) *Keyword privacy:* the cloud server could not identify the specific keywords in the query request and index or document collection by analyzing the statistical information like term frequency.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

IV. PROPOSED SEARCH SCHEMES

A. Search Module:

In our proposed scheme, Balanced M-way Search Tree (BMS) is used for perform searching. Let's suppose cloud server has received 'n' encrypted documents $D=\{d_1, d_2, \ldots, d_n\}$. So, if user want to search with 't' keywords, DO generate a secret trapdoor encrypted using CRSA. The trapdoor containing the encrypted keywords is sent as token to the server. Then the server uses this trapdoor to match the encrypted keywords in the index tree node. If match found, stores the pointer to that document in encrypted database. The following Algorithm 3 gives the stepwise information about how search is to be done on B-Tree.

1) *Algorithm 3: Search_query (root, trapdoor)*

- *Input:* root, trapdoor containing keyword to be searched.
- *Output:* pointer to the documents containing the keywords, NULL if non-exist.
- NODE_x = Disk_Read(root).
- if NODE_x is an index node
  - (a) If there is an object O in NODE_x such that O: key=keyword, return O:value.
  - (b) Find the child pointer x:child[i] whose key range contains key.
  - (c) Return Search_Query(NODE_x: child[i], key).
    Else if there is an object O in NODE_x such that O:key=keyword, return O:value.
    Otherwise, return NULL.
    End if.

B. Ranking Module:

In large databases, it is quite common that the keyword might be matching with more number of documents. It is very difficult to decrypt every document and go through all documents. Therefore there is a need to ranking the documents based on their relevance score to the keywords. In our scheme we used TF×IDF to rank the documents. The proposed schemes consist of following phases:

1) *Setup:* The proposed system initialized with this step. In this data owner generates secret key. DO generate n-bit random vector S and two n×n invertible matrices M_1, M_2. Hence the secret key SK is a form of three tuples as $SK=\{M_1, M_2, S\}$.
2) *Generate Index(DC,SK):* Data owner (DO) extracts keywords from document collection and creates a dictionary with keywords and then creates an index vector for each document. DO construct unencrypted index tree by calling BuildBMSIndexTree(D) algorithm. Now split the each index vector P_u of node u as two random vectors P_{u1} and P_{u2} using S-bit vector. If S[i]=0 then Pu1[i] and $P_{u2}[i]$ will be set to same value of $P_u[i]$. If S[i]=1 then $P_{u1}[i]$ and $P_{u2}[i]$ will be set to two random values whose sum is equal to $P_u[i]$. the encrypted index vector for the index tree is $C_u=\{M_1T.P_{u1}, M_2TP_2\}$.
3) *Trapdoor(W):* The interested keywords are entered by the user. It generates a bit vector Q based on keywords and synonyms d, if keywords available in d then set the IDF values of term to a specific dimension in Q respectively, otherwise set to 0. So, apply the same splitting technique which we have applied earlier to Q, divide Q into Q' and Q'' to provide search privacy, then encrypt it. The encrypted query vector is denoted as
   $TD=\{M_1\text{-}1Q', M_2\text{-}1Q''\}$.
4) *Query(TD,C_u):* The query is processed using encrypted keywords of the trapdoor TD and encrypted index vector C_u, score of the documents is stored in node u. The same process is applied to all nodes in index tree. The relevance score calculation of the document is as

$$
\begin{aligned}
\text{Score} &= C_u, T \\
&= \{M_1^T.P_u^1, M_2^T F_u^2\} \times \{M_1^{-1}Q', M_2^{-1}Q''\} \\
&= M_1^T.P_u 1.M_1^{-1} + M_2^T.P_u^2.M_2^{-1}Q'' \\
&= P_u^{1T}.M_1 \times M_1^{-1}.Q' + P_u^{2T}.M_2 \times M_2^{-1}.Q'' \\
&= P_u^{1T}Q' + P_u^{2T}Q'' \\
&= P_u^{T}\cdot Q
\end{aligned}
$$

The above used scheme is used for provide the security against known cipher text model. To prevent the cloud server from data access patterns.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

C. Dynamic Update Operations:

After insertion or deletion of document, need to synchronously update the index by updating the nodes using the document identifiers in the BMS index tree. The process is as follows:

For updating: $\{I'_s, c_i\} \leftarrow$ GenUpdateInfo(SK, Ts,i,updtype)

Here updtype denotes the insertion or deletion operations of document d_i. The notation Ts denotes the set of consisting of the tree nodes that needs to be changed during the update.

V. RESULTS

Fig 2. Searching files using secret key and filename after successful login.

Fig 3.Users gets ranked documents and decrypt files with their decryption keys sent to email.

VI. CONCLUSION AND FUTURE WORK

In this, our work uses CRSA asymmetric algorithm for encrypted data files and index tree based on B-tree. CRSA increases the data privacy by its commutative nature; data in a file can be updated dynamically without affecting the overall performance of searching on BMS tree. If encrypted data is modified (insert and delete), then re-encryption of whole data is not needed. This is desirable feature it reduces the computation time. The future work will be concentrate for better data security in multi-user model for performance improvement, because we cannot say every data user is honest and may leads to any security problems means they can distributes the decrypted documents and share secret key to unauthorized users.

REFERENCES

[1] D.X.Song, D.Wagner and A.Perrig, "Practical techniques for searches on encrypted data", in security and privacy, 2000.S&P 2000. Proceedings.2000 IEEE Symposium on. IEEE, 2000, PP.44-55.
[2] 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.
[3] R. Curtmola, J.Garay, S.Kamara, and R.Ostrovsky, "Searchable symmetric encryption improved definitions and efficient constructions", in proceedings of the 13th ACM 2006, pp.79-88.
[4] J. Li, Q. wang, C. Wa ng, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing", in INFOCOM, 2010 proceedings IEEE, 2010, pp.1-5.
[5] Ning Cao, Cong Wang, Ming Li, Kui Ren, Wenjiing Lou, "Privacy Preserving Multi-keyword Ranked Search over Encrypted Cloud data", Parallel and distributed system, IEEE Transactions on, vol.25, no.1, pp.222-233, jan 2014.
[6] C. Wang et al., "secure ranked keyword search over encrypted cloud data", proc.ICDCS'10, 2010.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

[7] B.wang, S.Yu, W.Lou andY.T.Hou, "Privacy-preserving multi-keyword fuzzy keyword search over encrypted data in cloud", in IEEE INFOCOM, 104.

[9] Zhangjie Fu et al, "Multi-keyword ranked search supporting synonym query over encrypted data in cloud computing", IEEE conference,2013.

[10] J. Katz., and F. Zhang, "An efficient public-key encryption supporting disjunctions, polynomial equations and inner products", in Advances in cryptology-EUROCRYPT 2008. Springer, 2008, pp. 146-162..

[11] N. Cao, C. Wang, M. Li. K. Ren and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", in IEEE INFOCOM, April 2011, pp.829-83.

[12] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking", in Proceedings of the 8th ACM SIGSAC symposium on information , computer and communications security. ACM, 2013, pp.71-82.

[13] C. Orencik, M. Kantarcioglu, and E. Savas, "A practical and secure multi-keyword search method over encrypted cloud data", in cloud computing (CLOUD), 2013 IEEE 6th international conference on. IEEE, 2013, pp.390-397.

[14] W. Zhang, S. Xiao, Y. Lin, T. Zhou and S. Zhou, "Secure ranked multi-keyword search for multiple data owners in cloud computing", in Dependable systems and networks(DSN), 2014 44th Annual IEEE/IFIP international conference on. IEEE, 2014. pp. 276-286.