International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 06 , June 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

MULTI-KEYWORD RANKED SEARCH MODEL FOR ENCRYPTED CLOUD DATA

Dr. D. Bujji Babu[1], K. Divya Vani[2], N. Sirisha[3]

[1,2,3] *MCA Dept., QIS College of Engineering and Technology, Ongole, Andhra Pradesh, (India)*

ABSTRACT

Now a days the popularity of cloud computing is increasing rapidly, So more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. To outsource the sensitive data it should be encrypted for privacy requirements ,which obsoletes data utilization like keyword-based document retrieval. In this paper, to support dynamic update operations like deletion and insertion of documents, we present a "Multi-keyword ranked search model for encrypted cloud data". Ranked search can enable quick search search for the most relevant data , sending back only the top-k most relevant documents can effectively decrease network traffic. Specially the widely used TF-IDF model are combined in index construction and query generation. Due the use of special tree-based index structure "Greedy Depth First search", the proposed scheme can achieve sub-linear search time and deal with deletion and insertion of documents flexibly. The secure KNN algorithm is utilized to encrypt the index and query vectors.

***Keywords*: Cloud Computing, Multi-Keyword Ranked Search, Searchable Encryption**

I INTRODUCTION

Now a day's cloud computing[10] has been considered as a new model in both individual and enterprise infrastructure, that can organize storage, huge resource of computing and enable users on-demand network[12] access to a shared pool of computing resources with great efficiency.

The individuals and enterprises are attracted by these appealing features of cloud services to outsource their data instead of purchasing software and hardware to manage the data. Outsourcing sensitive information(emails, personal health records etc) to remote servers to provide privacy to the data it should be encrypted, which is general approach to protect data confidentiality[7].

To address above problem, more practical special purpose solutions, such as searchable encryption(SE) schemes have made specific contributions in terms of efficiency, functionality and security. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search,[8],[3] multi-keyword Boolean search, ranked search, multi-keyword ranked search, etc. Recently, some dynamic schemes have been proposed to support inserting and deletion of operations on document collection. But only few schemes support efficient multi-keyword ranked search.

In this paper propose a secure tree based search scheme supports multi-keyword ranked search and dynamic operation on the document collection. Specially, the widely used vector space model along with TF(term frequency)-IDF(inverse document frequency) model are combined in the index construction and query

generation. To obtain high search efficiency, we construct a tree based index structure and propose a "Greedy Depth First Search" algorithm based on the index tree. The secure KNN algorithm is utilized to encrypt 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 scheme (BDMRS) in the known cipher text model, and enhanced dynamic multi-keyword ranked search scheme (EDMRS) in the known background model. Our contributions are

1) Searchable encryption scheme that supports both multi-keyword search and dynamic operation on document collection.

2) The proposed scheme can achieve higher search efficiency by executing our "Greedy Depth first search" algorithm.

II LITERATURE SURVEY

Due to the different cryptography primitives, Searchable encryption schemes can constructed using public key based cryptography[8],[9] (or) symmetric key based cryptography,[5][11]. The early works are based on single keyword Boolean search schemes, which are very simple in terms of functionality. Afterward, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search[2],[3],multi-keyword Boolean search[14],[18], ranked search [12],[6], and multi-keyword ranked search [16],[17],etc.

Multi-keyword Boolean search allows the users to input multiple query keywords to request suitable documents. Among these works , conjunctive keyword search schemes [7],[4] only return the documents that contain all the query keywords. Disjunctive keyword search schemes [9],[1] return all the documents that contain a subset of the query keywords. All these multi-keyword search schemes retrieve search results based on existence of keywords. Ranked search can enable quick search of the most relevant data.

Secure multi-keyword search[14] that supports similarity based ranking. The authors are constructed a searchable index tree based on vector space model TF-IDF to provide ranking results. In this paper proposed a secure multi-keyword search method which utilized local sensitive hash(LSH) functions to cluster similar documents. The LSH algorithm is suitable for similar search but can't provide exact ranking.

Thus , the SE schemes are expected to support the insertion and deletion of the documents. There are also several dynamic searchable encryption schemes. In the work of the each document is considered as sequence of fixed length keywords. In this paper Cash et al.[15] proposed dynamic searchable encryption scheme.

III METHODOLOGY

3.1 Notations and Preliminaries

- W- The set of keywords denoted as W={w1,w2,.....,wm}.
- m- The total number of keywords in W.
- Wq- The subset of W, representing the keywords in the query.
- F- The plaintext document collection, denoted as a collection of n documents F={f1,f2,...,fn}.
- n- The total number of documents in F.

- C- The encrypted document collection stored in the cloud server, denoted as C={c1,c,...,cn}.
- T- The unencrypted form of the index tree for the whole document collection F.
- I- The searchable encrypted tree index generated from T.
- Q- The query vector for keyword set Wq.
- TD- The encrypted form of Q, which is named as trapdoor for the search request.
- Du- The index vector stored in tree node u. u can be either a leaf node or an internal node of the tree.
- Iu- The encrypted form of Du.

3.1.1 Vector space model and relevance score function

Vector space model along with TF-IDF rule is widely used in plaintext information retrieval, which efficiently supports ranked multi-keyword search. Here, the term frequency(TF) is the number of times a given keyword appears within document, and inverse document frequency(IDF) is obtained through dividing the cardinality of document collection by the number of documents containing the keyword. In the vector space model each document is denoted as a vector and each query is also denoted as a vector Q. The dot product of the TF vector Du and IDF vector Q can be calculated to quantify the relevance between query and document. Following are the notations used in our relevance evaluation function:

- Nf, wi- The number of keyword wi in document f.
- N- The total number of documents.
- Nwi- The number of documents that contain keyword wi
- $TF'f, wi$- The TF value of wi in document f.
- $IDF'wi$- The IDF value of wi in the document collection.
- TFu, wi- The normalized TF value of keyword wi stored in the index vector Du.
- $IDFwi$- The normalized IDF value of keyword wi in document collection.

The relevance evaluation function is defined as:

$: RScore(Du, Q) = Du.Q = \sum_{wi \in W} TFu, wi \times IDFwi$ ----------(1)

If u is an internal node of the tree

$: TFu, wi = \frac{TF'f, wi}{\sqrt{\sum wi \in Wq\,(TF'f, wi)2}}$ ---------- (2)

where $TF'f, wi = 1 + lnNf, wi$. and in the search vector Q. $IDFwi$ is calculated as

$: IDFwi = \frac{IDF'wi}{\sqrt{\sum wi \in Wq\,(IDF'wi)2}}$ ----------(3)

where $IDF'wi = \ln(1 + \frac{N}{Nwi})$.

3.1.2 Keyword Balanced Binary tree

The KBB tree in our scheme is a dynamic data structure whose node stores a vector D. The elements of vector d are normalized TF values. Sometimes, we refer the vector D in the node u to Du for simplicity. The node u in our KBB tree is defined as follows.

: u= $< ID, D, Pl, Pr, FID >$ ---------- (4)

where ID denotes the identity of node u, Pl and Pr are respectively the pointers to the left and right child of node u. If u is a leaf node of the tree, FID stores the identity of a document, and D denotes a vector consisting TF values of the keywords to the document. If the node u is an internal node, FID set to null, D denotes a vector consisting of the TF values which is calculated as follows.

: D[i]= $\max\{u.Pl \rightarrow D[i], u.Pr \rightarrow D[i]\}, i = 1, \ldots, m$ ---------- (5)

The above notations and formulae's are developed by Zhihua Xia, Xinhui Wang.. [19] in "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data".

3.2 The system and Threat Models

The system model in this paper involves three different entities: data owner, data user and cloud server as illustrated in Fig. 1.

Architecture:

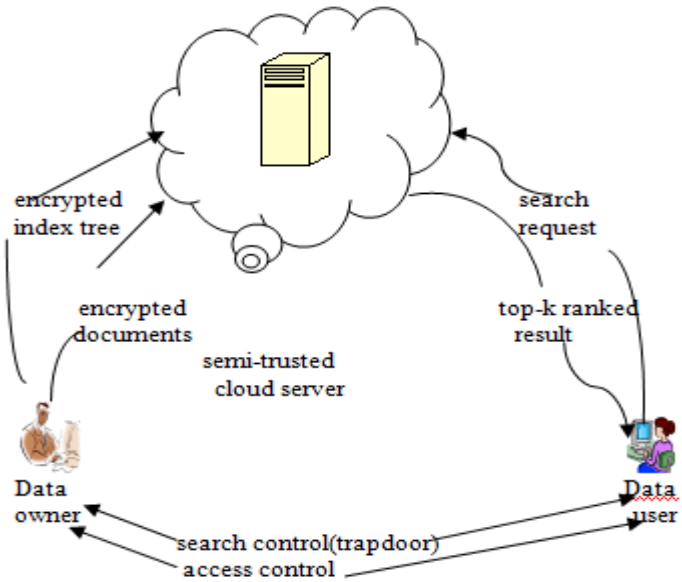


Fig.1.The architecture of ranked search over encrypted cloud data

3.2.1 Data Owner

Data Owner has a collection of documents F={f1,f2,...,fn}.Sensitive data has to be encrypted prior to outsourcing for data privacy. In our scheme, the data owner firstly builds a secure searchable tree index I from document collection F. and then generates an encrypted document collection and securely distributes the key information of trapdoor generation and document decryption to the authorized data users.

3.2.2 Data Users

Data users are authorized ones to access the documents of data owner. The authorized user can generate a trapdoor TD according to search control mechanisms to fetch k encrypted documents from cloud server. Then, the data user can decrypt the documents with the shared secret key.

3.2.3 Cloud server

Cloud server stores the encrypted document collection C and the encrypted searchable tree index I for data owner. Upon receiving the trapdoor TD from the data user, Upon receiving the trapdoor TD from the data user,

the cloud server executes search over the index tree I, and finally returns the corresponding collection of top-k ranked encrypted documents.

3.2.4 Known Cipher-text model

In this model, the cloud server only knows encrypted document collection C, the searchable tree index I, the search trapdoor submitted by the authorized user. That is, the cloud server can conduct cipher-text-only attack in this model.

3.2.5 known Background model

Compared with Known cipher text model, the cloud server in the stronger model is equipped with more knowledge, such as the term frequency statistics of document collection. This statistical information records how many documents are there for each term frequency(TF) of a specific record in whole document collection. The cloud server can conduct TF statistical attack to even identify certain keywords.

3.3 Design Goals

To enable secure, efficient, accurate and dynamic multi-keyword ranked search over outsourced encrypted cloud data under above models, our system has the following design goals.

3.3.1 Dynamic*:*

This scheme is not only multi-keyword query but also dynamic update on document collection.

3.3.2 Search efficiency:

The scheme aims to achieve sub-linear search efficiency by exploring a special tree based index and efficient search algorithm.

3.3.3 Privacy-Preserving:

The scheme is designed to prevent the cloud server from learning additional information about the document collection. The specific privacy requirements are

3.3.2.1 Index confidentiality and query confidentiality: This scheme is underlying plaintext information, TF values of keywords stored in the index and IDF values query keywords, should be protected from cloud server.

3.3.2.2 Trapdoor Unlink ability: The cloud server should not be able to determine whether the two encrypted queries generated from same search request.

3.3.2.3 Keyword privacy: The cloud server could not identify the specific keyword in query, index and document collection by analyzing the statistical information like term frequency.

IV USED SCHEMES

We firstly describe the unencrypted dynamic multi-keyword ranked search(UDMRS) scheme which is constructed on the basis of vector space model and KBB tree. Based on the UDMRS scheme, two secure search schemes (BDMRS and EDMRS) are constructed two threat models, respectively.

4.1 Index construction of UDMRS scheme

In the process of index construction , we first generate a tree node for each document in the collection. These nodes are the leaf nodes in the index tree. Then, the internal nodes are generated based on these leaf nodes. The formal construction process of the index is presented in Algorithm 1.The index tree built in here is a plaintext.

The following are some notations for Algorithm 1. The data structure of the tree node is defined as $< ID, D, Pl, Pr, FID >$, where ID for each tree node is generated through the function GenId().

- Current Node Set-The set of current processing nodes which have no parents. If the number of nodes is even, the cardinality of the set is denoted as 2h, else the cardinality is (2h+1).
- Temp Node Set- The set of newly generated nodes.

In the index, if Du[i]!=0 for an internal node there is at least one path from node u to some leaf node.

4.2 Search process of UDMRS scheme

The search process of this scheme is a recursive procedure upon the tree, named as "Greedy depth first search algorithm(GDFS)". We construct a result denoted as RList, whose element is defined as <RScore, FID>. Here RScore is Relevance score of the document $fFID$ to the query which is calculated according to formulae(1). The RList stores the k accessed documents with the largest relevance scores to the query.

Algorithm 1 BuildIndexTree(F)

Input: the document collection F = {f1,f2,...,fn} with the identifiers FID = {FID|FID = 1,2,...,n}.

Output: the index tree T

1: for each document $fFID$ in F do

2: Construct a leaf node u for $fFID$, with $u.ID = GenID()$, $u.Pl = u.Pr = null$, $u.FID = FID$, and D[i] $= TFfFID, wi$ for i = 1,...,m;—

3: Insert u to CurrentNodeSet;

4: end for

5: while the number of nodes in CurrentNodeSet is larger than 1 do

6: if the number of nodes in CurrentNodeSet is even, i.e. 2h then

7: for each pair of nodes u′ and u″ in CurrentNodeSet do

8: Generate a parent node u for u′ and u″, with $u.ID =$ $GenID()$, $u.Pl = u'$, $u.Pr = u''$, $u.FID = 0$ and $D[i] =$ $max\{u'.D[i], u''.D[i]\}$ for each i = 1,...,m;

9: Insert u to TempNodeSet;

10: end for

11: else

12: for each pair of nodes u′ and u″ of the former (2h−2) nodes in CurrentNodeSet do

13: Generate a parent node u for u′ and u″;

14: Insert u to TempNodeSet;

15: end for

16: Create a parent node u1 for the (2h−1)-th and 2h-th node, and then create a parent node u for u1 and the (2h + 1)-th node;

17: Insert u to TempNodeSet;

18: end if

19: Replace CurrentNodeSet with TempNodeSet and then clear TempNodeSet;

20: end while

21: return the only node left in CurrentNodeSet, namely, the root of index tree T;

Algorithm 2 GDFS(IndexTreeNode u)

1: if the node u is not a leaf node then
2: if RScore(Du,Q) > kthscore then
3: GDFS(u.hchild);
4: GDFS(u.lchild);
5: else
6: return
7: end if
8: else
9: if RScore(Du,Q) > kthscore then
10: Delete the element with the smallest relevance score from RList;
11: Insert a new element⟨RScore(Du,Q),u.FID⟩ and sort all the elements of RList;
12: end if
13: return
14: end if

kthscore- The smallest relevance score in current RList, which is initialized as 0.
hchild,lchild- The child node node of a tree node with higher,lower relevance score.
The above Algorithm1 BuildIndextree(F) and Algorthm2 GDFS are developed by Zhihua Xia, Xinhui Wang,.. [19] in "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data".

4.3 BDMRS scheme

Based on UDMRS scheme, we construct the basic dynamic multi-keyword ranked search(BDMRS) scheme by using the secure KNN algorithm. The BDMRS scheme is designed to achieve the goal of privacy preserving in the known cipher text model, and 4 algorithms included.

- $SK \leftarrow Setup()$ The secret key(SK) is generated from data owner. It including 1) a randomly generated m-bit vector. 2) two invertible matrices M1 and M2. Namely, SK={S,M1,M2}.
- $I \leftarrow GenIndex(F, SK)$ First the unencrypted index tree T is built on F by using $T \leftarrow BuildIndexTree(F)$. Secondly the data owner generates random vectors $\{Du', Du''\}$ for index vector Du in each node u. If s[i]=0, the two are equals to $Du[i]$; if s[i]=1, $Du'[i]$ and $Du''[i]$ will be set whose sum equals to $Du[i]$. Finally, the encrypted index tree I is built where the node u stores two encrypted index vectors $Iu = \{M1^T Du', M2^T Du''\}$
- $TD \leftarrow GenTrapdoor(Wq, SK)$ with keyword set Wq, the unencrypted query vector Q with length of m is generated. Finally the algorithm returns trapdoor $TD = \{M1^{-1}Q', M2^{-1}Q''\}$
- $RelevanceScore \leftarrow SRScore(Iu, TD)$ with the trapdoor TD, the cloud server computes the relevance score of the node u in the index tree I to the query. Encrypted vectors equals to unencrypted vectors.
  : $Iu.TD = Du.Q = RScore(Du, Q)$ ---------- (6)

4.3.1 Security analysis:

The three privacy requirements in BDMRS scheme are as follows.

4.3.1.1 Index Confidentiality and Query Confidentiality: The BDMRS scheme is resilient against cipher text only attack and index confidentiality and query confidentiality are well protected.

4.3.1.2 Query Unlink ability: The trapdoor of query vector is generated from a random splitting operation, which means that the same search request will be transferred into different query trapdoors, and query unlink ability is protected.

4.3.1.3 Keyword Privacy: The keyword privacy is protected in the known cipher text model. But in known background model, the cloud server is supposed to have more knowledge, such as the term frequency of the keywords. The BDMRS scheme can't resist the TF statistical attack in the known background model.

4.4 EDMRS scheme

The BDMRS scheme can protect the index and query confidentiality in the known cipher text model. In the addition, known background model, it is possible for the cloud server to identify the keyword as the normalized TF distribution of the keyword can be exactly obtained from final calculated relevance scores. The primary cause is that the relevance score is calculated from Iu and TD is equal to the Du and Q.

- $SK \leftarrow Setup()$ In this algorithm to set the secret vectors S as m-bit vector and set M1 and M2 are invertible matrices.
- $I \leftarrow GenIndex(F, Sk)$ Before encrypting the index vector Du, we extend the vector Du to be (m+m') dimensional vector.
- $TD \leftarrow GenTrapdoor(Wq, Sk)$ The query vector Q is extended to be a (m+m')
- $Relevancescore \leftarrow SRScore(Iu, TD)$ The Relevance score for index vector Iu equal to $Du.Q + \sum \epsilon_g$

4.4.1security analysis:

The security of EDMRS scheme is analyzed according to the three predefined privacy requirements in the design goals:

4.4.1.1 *Index and Query confidentiality:*

Inherited from BDMRS scheme, the EDMRS scheme can protect index and query confidentiality in the known background model.

4.4.1.2 Query Unlink ability: The same search requests will generate different query vectors and receive different relevance score calculations by introducing random value ϵ. Thus, Query unlink ability is protected better.

4.4.1.3 *Keyword privacy*: The BDMRS scheme cannot resist TF statistical attack in known background model. The cloud server is able to identify keywords through analyzing the TF distribution histogram.

4.5 Dynamic update operation of DMRS:

The index of DMRS scheme is designed as a balanced binary tree, the dynamic operation is carried out by updating the nodes in the index tree. The specific process is presented as follows.

- $\{Is', Ci\} \leftarrow GenUpdateInfo(SK, Ts, i, updtype))$ This algorithm generates the update information $\{Is', Ci\}$ which will be sent to cloud server. Here the notation $updtype \in \{Ins, Del\}$ denotes either insertion or deletion for the document fi. Ts denotes tree nodes.

  - If *updtype* is equal Del, the data owner deletes from the sub tree the leaf node that stores the document identity i and updates the vector D of other nodes in sub tree Ts, so as to generate the update sub tree Ts'

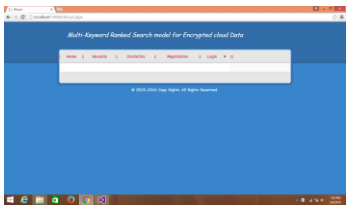
  - If *updtype* is equal to Ins, the data owner generates tree node u =< $GenID$ (), $D, null, null, i >$ for the document fi.

- $\{I', C'\} \leftarrow Update(I, C, updtype, Is', Ci)$ In this algorithm, cloud server replaces the corresponding sub tree Is with Is', so generate a new index tree I'.

  The above formulae's are developed by Zhihua Xia, Xinhui Wang,.. [19] in "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data"**.**

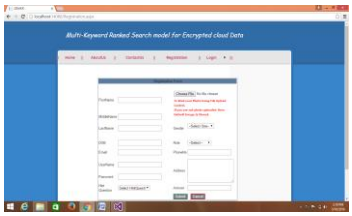
V RESULTS

For the implementation of this project we used Visual studio 2013 and SQL server 2014 technologies and we present the output screens are as follows.



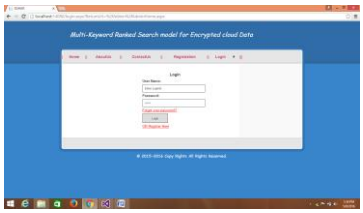
(A).Home Page

The above screen displays home page. In this home ,about us, contact us, registration, and login buttons will be there. We can click the registration button then displays a form



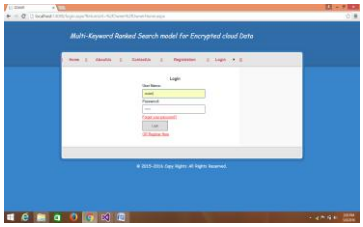
(B).Registration Form

The above screen describes the registraion form. Fill all the columns and click the submit button.Then user and data owner will be successfully registered. We can click the admin button it displays form.



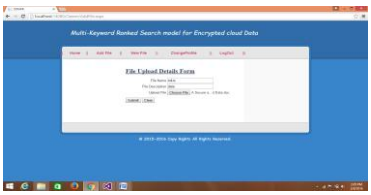
(C). Admin Page

The above screen is the admin login form to be create login form two textboxes is created to enter username and password to click the login button then displays a admin home page. It accepts the roles of user and owner.



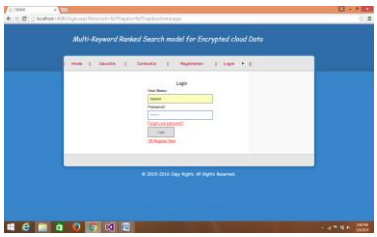
(D).Data owner login page

The above screen is the owner login form. In this enter the username and password. we can click login then display a form.



(E).Add File

This screen shows the file upload details form. We fill the details of file name, description and click the submit button then file will be uploaded and generated by the trapdoor successfully.

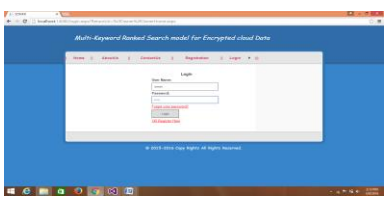


(F).Trapdoor login page

The above screen is the trapdoor login form we can enter the user name and password and click the login form.

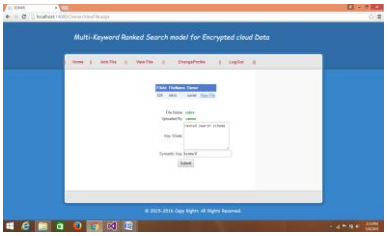
(G).New request file in Trapdoor

The above screen is the new request file form we can click the execute button then display the upload file duplicate words will be open. We can enter the file expected words in the box and click forward button. Then log out the form.



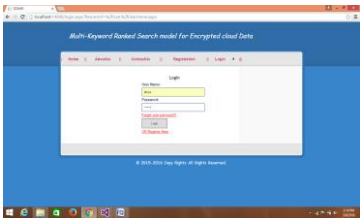
(H).Owner login page

The above screen show the owner login form enter username and password and click login then display a form, click on view file button.



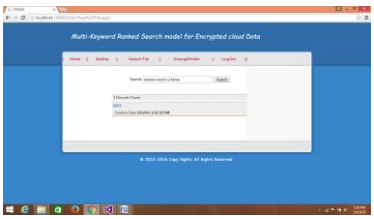
(I)Enter keywords in view file

The above screen is the view file data and enter the multiple keywords and Symantec key will be generated. We click on submit button then file will be accepted.



(J).User login page

The above screen is the user login form enter user name and password click login button then open a user login form.



(K).Search file

The above screen is the search file form. we can enter the multiple keywords in the textbox and click search button. If we down load the file get the key we click on get key button then get the key and then download.

(L).Output file

The above screen is the output of the file will be open.

VI CONCLUSION AND FUTURE WORK

In this paper, we present a secure, efficient and dynamic search scheme is presented, which supports the dynamic deletion and insertion of documents. We construct a special index tree , and propose a "Greedy Depth-

first search" as shown in Algorithm2 to obtain better efficiency than linear search. The secure KNN algorithm is used to protected against two threat models.

In the proposed scheme , the data owner is responsible for updating the information and sending them to cloud server. It could be a meaningful but difficult future work to design a dynamic searchable encryption scheme whose updating operation can be completed by cloud server only. Actually In multi-user scheme many secure challenges will be there. Firstly, all the users usually keep the same secure key for trapdoor generation in a symmetric SE scheme. If it is needed to revoke a user in this scheme, we need to rebuild the index and distribute the new secure keys to all the authorized users. Secondly SE schemes usually assume that all the data users are trustworthy. In the future works , we will try to improve the SE scheme to handle these challenging problems.

VII ACKNOWLEDGEMENTS

The satisfaction that accomplishes the successful completion of the project would be incomplete without the mention of the people who made it possible. I consider it my privilege to express my gratitude and respect to all those who guided and inspired in the successful completion of my work.

I am grateful to our college secretary and correspondent Sri. N.S. Kalyan Chakravarthi and the president of this SNES Sri. N. Nageswara Rao, for providing me an opportunity to utilize the infrastructural facilities and computational facilities at **QIS College Of Engineering and Technology, Ongole** and allowing availing the entire faculty in the college.

REFERENCES

[1] B. Zhang and F. Zhang, "*An efficient public key encryption with conjunctive-subset keywords search,*" Journal of Network and Computer Applications, *vol. 34, no. 1, pp. 262–267, 2011*.

[2] 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.

[3] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "*Efficient similarity search over encrypted data," in Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012, pp. 1156–1167.*

[4] Y. H. Hwang and P. J. Lee, "*Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Proceedings of the First international conference on Pairing-Based Cryptography.* Springer-Verlag, *2007, pp. 2–22.*

[5] E.-J. Goh et al., *"Secure indexes*." IACR Cryptology ePrint Archive, *vol. 2003, p. 216, 2003.*

[6] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+ r: Topk retrieval from a confidential index," in Proceedings of the12th international Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009, pp. 439–449.

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

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

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

[10] D.Bujji Babu.,etl., "*A New Cloud based and Cyclic approach to Solve a classification problem"*. International Conference on Emerging Trends in Engineering and Technology (ICETET 2013) International conference Program Dec 7-8, 2013 Patong Beach, Phuket ISBN:978-93-5137-024-6.

[11] D.Bujji Babu.,etl.,"*Cryptanalysis of a Fistel Type Block Cipher by Feed Farward Neural Network using Right Sigmodial Signals"*. International Journal of Soft Computing 4(3):131-135,2009. ISSN:1816-9503.

[12] D.Bujji Babu.,etl.,"*A Neural Network Model for Computer Network Security"*. Advances in Computation Sciences and Technology. Research India Publications. 2(2009)pp 179-186 ISSN 0973-6107.

[13] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and Network Security. Springer, 2004, pp. 31–45.

[14] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Advances Cryptology–EUROCRYPT 2008. Springer, 2008, pp. 146–162.

[15] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M.Wu, and D.W.Oard, "Confidentiality-preserving rank-ordered search," in Proceedings of the 2007 ACM workshop on Storage security and survivability. ACM, 2007, pp. 7–12.

[16] 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. 829–837.

[17] 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 similarity-based ranking," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013, pp. 71–82.

[18] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M.Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in Proc. of NDSS, vol. 14, 2014.

[19] Zhihua xia, X.Wang, X.Sun, Q.Wang "A Secure and Dynamic multi-keyword ranked keyword search scheme over encrypted cloud data," in 2015.