ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

The Time Efficient Privacy-Preserving Multi-Keyword Ranked Search more Encrypted Cloud Data

[1] Rajaram Jatothu,[2]Dr.R.Vivekanandam
[1] Dept of CSE,Vignan Institute of Technology and Science,Deshmukhi,Hyderabad.
[2] Professor, Director in Muthayammal Eng. College, Rasipuram, Tamilnadu.

Abstract: In cloud environment data privacy is very important concern. Hence while deploying any data on cloud particular needs higher security with privacy. Hence important or personal data have to be encrypted before deploying that data on cloud. When particular deploys data and want to retrieve some data then he/she has to go under many search techniques to find the relevant document according to search query. With consideration of privacy and of economic concerns most individuals tend to deploy data on cloud server or deploying data from local to commercial cloud. But while deploying data security and privacy are most important Hence in this paper we are defining first time mutli-keyword ranked search with preserving privacy along with that we have to achieve time efficiency, for that we are using multithreading concept. We first propose a basic idea for the MRSE based on secure inner product computation to efficiently achieve multi- keyword ranked search and then give two significantly improved MRSE schemes to achieve various privacy requirements.

Keywords: Encrypted Data Search,Cloud Service Providers, Cloud Storage, Ranked Search.

INTRODUCTION

Cloud computing has large scope now a days, particular individuals and enterprises outsource their important data on cloud for better economic saving. While putting that data on to the cloud one needs greater security, because individuals may have their personal data such as emails, their photos, personal health records, also enterprises may have their personal data like employees personal information, patent files. While putting data on cloud another important concern comes is their privacy. Hence while putting data on to cloud one needs privacy and security.

In previous papers, there was concept of single keyword ranked search but in this paper we are achieving privacy with multi-keyword ranked search with time efficiency. While deploying any data on cloud particular needs privacy, and for that, that data to be encrypted before deploying on cloud. When this data is deployed on cloud and again after downloading that data we have to decrypt that data,but as band width cost of cloud is high hence its impractical to do this. Also one don‘t need particular storage if we deploy data on cloud. In this paper for the first time we solve the problem of multi-keyword ranked search of encrypted data while preserving privacy with time efficiency. For this we used firstly co-ordinate matching i.e. as many Searches as possible to retrieve particular document according to their relevance. We also used inner productivity to find out similarity measure between multiple keywords entered in search query with existing document to achieve ranking for retrieving documents.
For ranked search to utilize out-sourced cloud data, our system design should simultaneously achieve security and performance guarantees, privacy and time efficiency as follows.

Multi-keyword Ranked Search:
it designs search schemes which allow multi-keyword query. It compares similarity between entered query keywords and keyword existing in the document to retrieve particular document and returns result, instead of returning undifferentiated results.

Privacy-Preserving:
For achieving privacy we have to keep it in encrypted format on cloud, as bandwidth cost is high in cloud it will be good to store that data on cloud, rather than downloading and decrypting that data.
Time-Efficiency:
Privacy should be achieved with low communication and computation overhead and time efficiency should be achieved with multithreading that is implementing taskparallely.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

II. LITERATURE SURVEY

A. Secured Multi-keyword Ranked Search over Encrypted Cloud Data[1]:

Considering the large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. In this paper, challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing. Among various multi-keyword semantics,choose the efficient similarity measure of coordinate matching, i.e., as many matches as possible, to capture the relevance of data documents to the search query. Further use inner product similarity to quantitatively evaluate such similarity measure. First proposed a basic idea for the MRSE based on secure inner product computation to efficiently achieve multi-keyword ranked search and then give two significantly improved MRSE schemes to achieve various privacy requirements.The main limitation of this paper was, the users identity (ID) is not kept hidden. Due to this, whoever puts the
data on Cloud Service Provider was known. This may be risky in some situations where confidentiality of data needs to be maintained. Hence, this drawback is overcome in the proposed system.

B. Privacy Preserving Keyword Searches on Remote Encrypted Data [2]:

Let us consider an example: a user named ram wants to store his files on remote server s, and he has to preserve privacy for that he will do encryption to achieve security so that keywords will also be to them-selves secret and not to endanger the security of the remotely stored files.
For example, ram may want to store old e-mail messages encrypted on a server managed by another large vendor like yahoo, and later retrieve certain messages while travelling with a mobile device. In [2], solutions for this problem under well-defined security requirements can be given. The schemes are efficient as no public-key cryptosystem is involved. In this, use ram can submit new files which are secure against previous queries but still searchable against future queries.

C. Efficient and Secure Multi-Keyword Search on Encrypted Cloud Data [4]:

Users are not aware of,encrypted data that deployedon cloud, hence when user enters keywords in the search query all files will be retrieved, furthermore user has to process that encrypted files in order to obtain matching files of their interest. Here in this case network traffic will be increased while ranking most relevant documents, in order of keyword search query. This all phenomenon will be totally undesirable in in pay-as-you –use cloud paradigm.This paper has defined and solved the problem of effective yet secure ranked keyword search over encrypted cloud data [4].

This paper did first time practical deployment of privacy preserving multi-keyword search,i.e. it has defined and solved challenging problem of privacy preserving multi–keyword ranked search over encrypted cloud data. The algorithm used for proposed ranking systemgives best results according to the keywords entered in the search query. Hence we referred thisalgorithm in order to get relevant results along with security of data on cloud.

D. Providing Privacy Preserving in Cloud Computing [5]:

Privacy is an important issue while preserving data in large era. As there are large no of individuals and enterprises keeps their important data on cloud, privacy becomes very big issuein cloud computing. As no of individuals and enterprises post there data, they also keep concerns about privacy and security of their data.
Hence as of maintaining enterprises and individuals this paper tells that, privacy is very important thing that has to be taken into consideration. From this paper we referred privacy preserving of data. Drawback of this paper is, it doesn‘t allow indexed search and also doesn‘t hide users identity i.e. who is deploying data and who is retrieving data. These two drawbacks are overcome in our proposed system.

E. Efficient Fuzzy Keyword Search over Encrypted Data in Cloud Computing [7]:

This paper solves effective fuzzy keyword search over encrypted cloud datamaintaining keyword privacy [7].
We have taken formulation from this paper to achieve multi-keyword ranked search more encrypted cloud data along with privacy preserving. In this paper design is made such that one should get reliable performance with efficient system usability.

F. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing[9]:

Usually there were papers who, preserved privacy while searching and ranking keywords along with their relevance. This paper approaches to obtain fine grained

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

data, scalability, and differentiated results. This paper addresses access policies based on data and also it delegates computations involved in data access. It utilizes single authenticator and random masking to guarantee that the any third party administrator would not learn any knowledge about the data content stored on the cloud server during the retrieving data, which eliminates the burden of cloud.

III.EXISTING SYSTEM

As shown in the fig system consist of following three entities

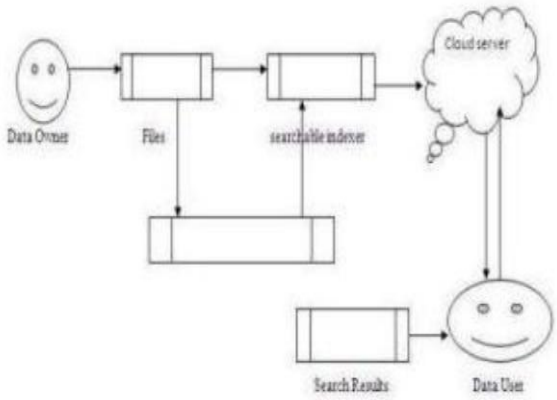

Fig. 1. Search on encrypted data

Data Owner-He has a collection of documents. He want to store these documents on cloud storage. Data owner perform preprocessing (i.e. removal of _ing' from verb, and tokenizing all Words), index construction, encryption and Upload of documents.

Data Users –Users wants to access these files. Using search keywords and secure key a trapdoor TD is generated. Trapdoor TD is used to search. After search processes, the cloud server sends most relevant k files to user. The retrieved documents are decrypted using secrete key.

Cloud Server- Upon receiving request form user, the cloud server searches the index tree and sends back top k relevant files to user.

Disadvantages of existing system
Supports only identical keyword search, semantic search is not supported. Data owner has to perform all pre-processing tasks Index encryption is sequential.

IV. PROPOSED SYSTEM

As shown in fig the proposed system consist of four entities:- Data Owner, Web Application, Data User and Cloud Server. In this system the load data owner is reduced by adding one user application. Data pre-processing, encryption, decryption all task are done by user application System module consists of three modules as data preprocessing i.e. tokenization of words, construction of index tree required for searching, and query processing module. Data preprocessing module performs various preprocessing functions on entire data set and generate list of tokenized terms. Construction of index tree module creates index tree from we have to search particular keyword by using term frequency values (TF) values of keyword from each documents. Query processing module process the query terms and search more index tree. The similarity between the terms and document is calculated and set ranked relevant documents are returned to the user.
Mathematical Model

We consider set theory: Suppose there exists system s,
We have to apply set theory on this system s, To deal with S,
S={s,e,X,Y,Fcf,FS,SS};

Notation: s=start state
CF=set of noriginal text files, CF={f|f1,f2,f3,……fn}

Fcf={fl,fs,fst}

Fcf=is set of functions i.e. fl=function for lexical analysis, fs=is function for stopword removal
fst=is function for stemming

Fcf={ CF }→{T}
{T}=is the list of terms or keywords generated after preprocessing. The weight of each term is calculated usingfollowing equation
Weight=tf∗idf

Wheretfis a term frequency of term and idfis the inversedocument frequency of term

D-Setofmkeyworddictionary, D={D|D1,D2,D3,...Dm} after preprocessing

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

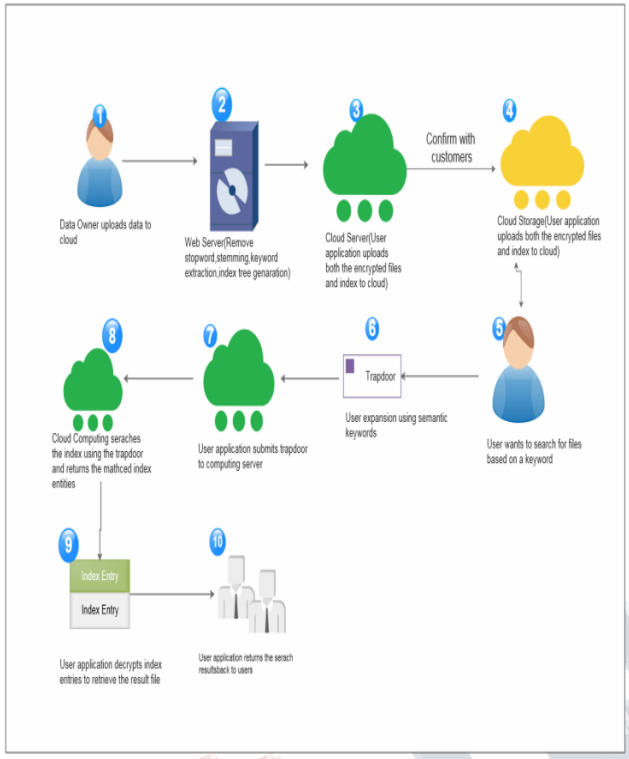

Fig. 2. Proposed Architectural Design

I - Searchable Keyword balanced binary(KBB) Index tree built form all files DC.
I - Encrypted index tree generated from I.
C- Set of n chipper text files after encryption, C={c|c1,c2,..,cn}

X = Input of the system
Here X is the search request or query entered by the user.
X=Q;
r - No. of expected relevant documents. For the query same document preprocessing functions are applied.

Y = Output of the system Y⊆CF={d1,d2,...,dk}
SS= Success state

FS=Failure state when no matching documents found on cloud with search query

Fcf= Functions of the system
={
Preproecssing(),generate _Index_Tree(),encryption(), Upload_doc_indextree_to_cloudserver(),genQuery(),search()}

V. SYSTEM ARCHITECTURE

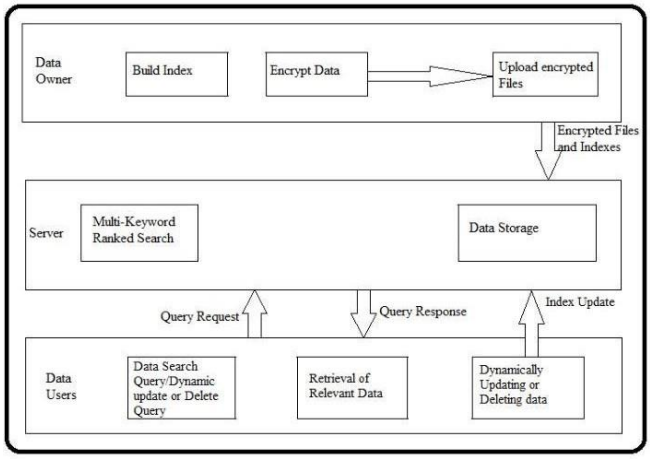

Fig. 3. System Architecture

1. Data owner will do all initial tasks, like encrypting files and then data owner will get keywords required for search query
2. Data owner will upload encrypted files, also keywords for search query encrypted and uploaded to server
3. Index generation occurs with build tree index algorithm
4. Only data owners can access that encrypted files End users can search for query, but doesn't have access to see encrypted files, and one can get permission from data owner.
5. When particular end user wanted to see contents of the file, he can send request to data owner
6. Data owner send the private key to the end users who did request through their mail, so that one can have access on contents of file.

The system consists of 4 main implementation steps:-

1. SK Setup (): Here system set the secret vector S as an m- bit vector.set M1 and M2 are (m + m') (m + m') invertible matrices, where m is the number of assumed terms.
2. Generate Index (F; SK): Before encrypting index vector Du, extend the vector Du to be (m+m') - dimensional vector. Each extended element Du [m+ j], j = 1... m', is set as a random number E (j).

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

3. TD Generate Trapdoor (Wq; SK): Query vector Q is extended to be (m + m') - dimensional vector. Among the extended elements, a number of m elements are randomly chosen to set as 1, and the rest are set as 0.

Relevance Score SRS core(Iu; TD): After the execution of relevance evaluation by cloud server, the final relevance score for index vector Iu equals to Du:Q + E(v); wherev j Q[m + j] = 1

VI.PROBLEM FORMULATION:

Setup after extracting the distinct keywords set W from the document collection F, data owner randomly generates a (n + 1)-bit vector as S and two (n + 1) (n + 1) invertible matrices $\{M_1, M_2\}$. The secret key SK is in the form of a 3-tuple as $\{S,M_1,M_2\}$.Build Index (F ,SK) Data owner generates a binary data vector D_i for every document F_i, where each bit $D_i[j]$ represents the existence of the corresponding key-word W_j in that document. Subsequently, everysubindex I_i is generated by applying dimension extending, splitting and encrypting procedures on D_i. These procedures are similar with those above except that the $(n+1)^{th}$ entry in Di is set to 1 during dimension extending. Finally,

sub index $I_i = \{M^T D^1, M^T D^{n}\}$ is built for every encrypted document C_i on cloud server.

Trapdoor ($\widetilde{W}$) with t keywords of interest in $\widetilde{W}$ as input, one binary vector Q is first generated Where each bit Q[j] indicates whether $W_j \in \widetilde{W}$ is true or false. Q is then scaled by a random number r ≠ 0 as rQ, and to a (n + 1)-dimension vector as $\vec{Q}$= (eQ, t) where t is another random number. After applying the same splitting and encrypting processes as above, the trapdoor

$T_{\widetilde{W}}$ is generated as $\{M_1^{-1}\vec{Q}, M_2^{-1}\vec{D''}\}$.

Query ($T_{\widetilde{W}}$, k, I) with the trapdoor $T_{\widetilde{W}}$, cloud server computes the similarity scores of each document F_i as in equation 1. W_{LOG}, we assume r > 0. After sorting all scores , cloud server returns the top k ranked id listFw

Similarity Score can be calculated as:

$$I_i.T_{\widetilde{W}} = \{M_1^T\vec{D'}_i, M_2^T\vec{D''}_i\}.\{M_1^{-1}\vec{Q'}, M_2^{-1}\vec{D''}\} \dots (1)$$
$$= \vec{D'}_i.\vec{q'} + \vec{D''}_i.\vec{q''} = \vec{D_i}.\vec{Q} = r(D_i.Q) + t$$

F - the plain-text document collection, denoted as a set of m data documents F = (F_1, $F_{2,\dots,}F_m$).
C - the encrypted document collection stored in cloud server, denoted as C = (C_1,C_2,…..,C_m).
W - the distinct keywords extracted from document collection F , denoted as W = (W_1,$W_{2,}$……., W_n).

I - the searchable index associated with C, denoted as (I_1, I_2,…..,I_m) where each sub index I_i is build for F_i.

W the subset of W , representing the keywords in a search request, denoted as Wf = (W_{j1}, W_{j2}…….W_{jt}).
T w. the trapdoor for the search request Wf.
Fw - the ranked id list of all documents according to their similarity with w.

VII.MATHEMATICAL MODEL

Set Theory:-
Let S be the system to perform Multi Keyword Ranked Search over the Encrypted Cloud Data. S= I, O, F, Fail, Success
Where, Inputs I = {I1, I2, I3,I4,I5} I1= User Login,
I2= Extract Keywords,
I3= Encryption of data using public key, I4= File and Keywords stored on cloud,
I5= Search over the cloud data using the keywords, I6= Decryption of data
Where, O is an Output Output(O)={O1,O2,O3,O4,o5,o6}
O1=Account Created,
O2= Keyword is Extracted in Encrypted form, O3= Keywords are used for Encryption,
O4= Updation in the index, O5= Encrypted data,
O6= Extraction of the original data.
Where, Functions set F
F1 Setup,
F2KeyGe n,
F3Encryp t, F4 Upload,
F5 Keyword Search, F6 Decrypt

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Failure Conditions: In this case, the search is not successful i.e., the user is not able to find the required document because there exists no keyword matching to the required query as required by the user.
Success Conditions: In this case, the search is successful i.e., the user is able to find the required document as the keyword exists in the index; the document is retrieved by the user.

Algorithms

Algorithm: Build tree-Index
Input: Document collection CF = {f1, f2,…..fn} with the identifiers ID = ID—ID = 1; 2... n.

Output: Index tree T
1) for each document/text file fID in F do
2) Construct leaf node v for each file fID, with v.ID = GenID (),and F[i] = TF*fIDi.eith index of any file.;

1) Now v becomes result node, so add v to Current NodeSet;
2)while number of nodes in Current NodeSet is greater than 1 do
3) if number of nodes in Current NodeSet is even, i.e. 2u then
4)while number of nodes in CurrentNodeSet is greater than 1 do
5)if number of nodes in CurrentNodeSet is even, i.e. 2u then for each pair of nodes v‘ and v‖ in CurrentNodeSet do
6) Generate parent node v for v‘ and v‖, with v.GID = GenID(), v.Pl = v‘, v.Pr = v‖, v.ID = 0 and D[i] = maxfv‘.D[i]; v‖.D[i]g for each i = 1,…..,m;
7) Add v to TempNodeSet.
8) else
10) for each pair of nodes v‘ and v‖ of former (2u -
11) nodes in CurrentNodeSet do
12) Create parent node v for v‘ and v‖;
13)Add v to TempNodeSet;
14) end for Generate parent node v1 for the (2u - 1)-th and 2u-tu node, and then generate parent node v for v1 and the (2u + 1)-tu node;
15) Add v to Temp Node Set;
16) end if
17) Replace Current Node Set with Temp Node- Set and then clear Temp Node Set;
18) end while
return only node left in Current Node Set, namely, root of index tree RT ;

VIII.RESULTS

The implementation of the proposed scheme is done using Java in Windows 7 operating system and tests its efficiency. The tests include Search precision on different privacy level. The experimental results are produced with an Intel Core(TM) i5 Processor.

The search precision of this system is affected by the assumed keywords in proposed method. Here, the precision is defined as that in [5]:

Pk = k‘/k,

where k‘ = the number of real k documents as top ranked in the retrieved k documents.

If a minor standard deviation is set for the random variable v, this technique is supposed to obtain larger precision, and vice versa.

The results are shown in fig 4. As said earlier here phantom, assumed terms are added to the index vector to change the relevance score calculation, so that the cloud server cannot detect keywords by checking the TF(term frequency) distributions of special keywords. Here, we evaluated the uncertainty of the relevance score by ―rank privacy‖, which is defined as:

$$P\,{}^{0}k = jr_i \qquad r\,{}^{0}j=k\ ;$$

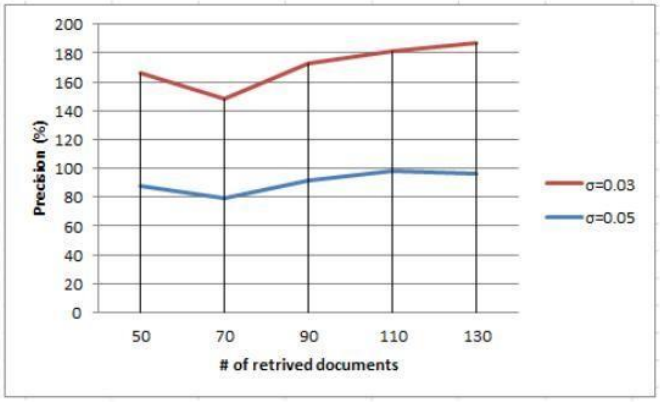

Fig. 4.Precision of searches with different standard deviation

Where ri is the rank number of document in the retrieved k documents as top documents, and r’ i is its real rank number in

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

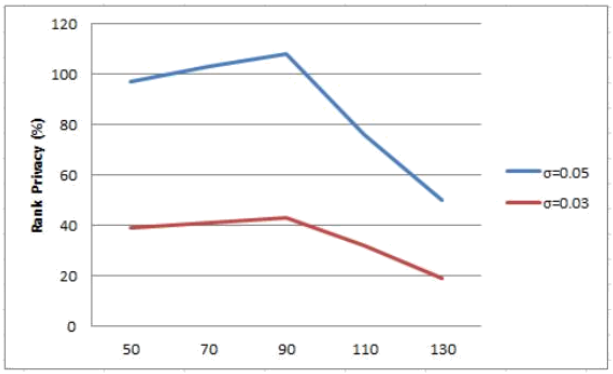

Fig. 5. Rank Privacy of searches with different standard deviation

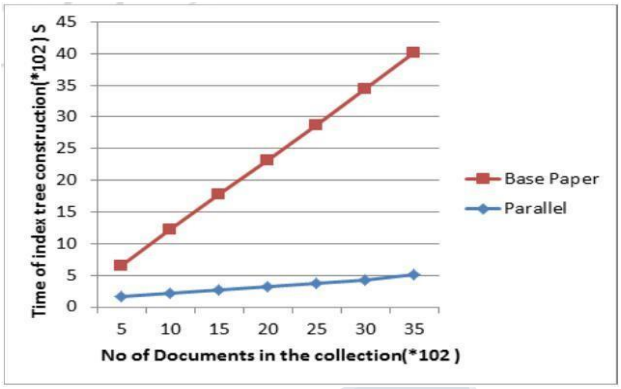

Fig. 6. Index tree construction cost for different sizes of document collection with the fixed dictionary, m=4000

the whole ranked results. The larger rank privacy denotes the higher security of the scheme, which is shown in Fig. 5. Here, data users can accomplish different requirements on search precision and privacy by adjusting the standard deviation.

Here the comparison is with a recent work, which achieves good search efficiency. The previous scheme retrieves the search results through exact calculation of document vec-tor and query vector. Thus, top-k search precision of the previous scheme is 91%. But as a similarity-based multi-keyword ranked search scheme, the previous scheme suffers from precision. The average precision of this method is 86%.

No of documents in the collection(10)	Time of index tree construction(10)^{2}s	
	Base Paper (Sequential)	Proposed System (Parallel)
5	5	1.59
10	10	2.15
15	15	2.72
20	20	3.23
25	25	3.78
30	30	4.34
35	35	4.91

IX.CONCLUSION

In this paper, we define and solve the problem of time efficient multi-keyword ranked search more encrypted cloud data, and establish a variety of privacy requirement. The short aim, MRSE using secure inner product computation. Then we give two significantly improved MRSE schemes using multithreading concepts while providing task to multiple threads rather than depending on single thread to yield time efficient result to achieve various privacy requirements in two different threat models.

X.FUTURE WORK

As our future work, we will explore other multi-keyword semantics more encrypted data; we can achieve ranking more than existing rank order in search result and privacy using the different algorithms with treading concepts.

REFERENCES

1. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, ―Privacy- Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data‖, Proc. IEEE INFOCOM, pp. 829-837, Apr, 2011.

2. L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, ―A Break in the Clouds: Towards a Cloud Definition‖, ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50-55, 2009.

3. N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, ―LT Codes- Based Secure and Reliable Cloud Storage Service‖, Proc. IEEE INFOCOM, pp. 693-701, 2012.

4. S. Kamara and K. Lauter, ―Cryptographic Cloud Storage‖, Proc. 14th Intl Conf. Financial Cryptograpy and Data Security, Jan. 2010.

5. A. Singhal, ―Modern Information Retrieval: A Brief Overview‖, IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35- 43, Mar. 2001.

6. I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishing, May 1999.

7. .D. Song, D. Wagner, and A. Perrig, ―Practical Techniques for Searches on Encrypted Data‖, Proc. IEEE Symp. Security and Privacy, 2000.

8. E.-J. Goh, ―Secure Indexes‖, Cryptology ePrint Archive, http:// eprint.iacr.org/2003/216. 2003.

9. Y.-C. Chang and M. Mitzenmacher, ―Privacy Preserving Keyword Searches on Remote Encrypted Data‖, Proc. Third Intl Conf. Applied Cryptography and Network Security, 2005.

10. R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, ―Searchable Symmetric Encryption: Improved Definitions and Efficient Construc-tions‖, Proc. 13th ACM Conf. Computer and Comm. Security (CCS 06), 2006.