iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30517

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

Multikeyword Rank Search Scheme for Unindexed Encrypted Cloud Data

Vaishali B Bambode[1], Dr. A A Bardekar[2]
[1, 2]*Computer of Science & Engineering Department, SIPNA College of Engineering*

Abstract: Due to the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which no longer support data utilization like keyword-based document retrieval. In this project, we present a secure multi keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations insertion and updating of documents. Specifically, We construct a special tree-based index structure and propose a "Greedy Depth-first Search" algorithm to provide efficient multi-keyword ranked search. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score i.e keyword weitage calculation between encrypted index and query vectors. In order to calculate the TF value of the search keyword we use a pattern matching algorithm which indicates the occurrence of that particular keyword in a file. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the insertion and updating of documents flexibly.

Index Terms: Cloud Computing, Multi keyword rank search scheme ,TF ,KNN algorithm, Greedy DFS algorithm

I. INTRODUCTION

Cloud computing has been emerged as a new model of IT infrastructure, which helps to organize huge resource of computing, storage and applications, and enable users to enjoy convenient and on demand network access to a shared pool of computing resources with great efficiency and minimal economic overhead . Because of these appealing features of cloud computing ,both individuals and enterprises are motivated to outsource their data to the cloud. Despite of various advantages of cloud computing services ,outsourcing sensitive information like e-mails, personal health records ,government data or documents to remote servers have always privacy concerns. The cloud service providers (CSPs) that keep the data for users may access users sensitive information without authorization. A general approach to protect the data confidentiality is to encrypt the data before outsourcing.

Multi-keyword ranked search over encrypted cloud data (MRSE) was introduced in 2014 by N. Cao et al. The main idea of this scheme was to allow users on search request and return documents with semantic multiple keywords . In order to secure and get the most relevant results retrieval, MRSE was adapted from secure k-nearest neighbor (kNN) technique to select the k nearest database records between database record and query vector. Secure inner product computation was adopted in order to set strict privacy requirement to ensure secrecy of cloud communication .

Recently, some dynamic schemes have been proposed to support inserting and updating operations on document collection. It is highly possible that the data owners need to update their data on the cloud server. But few of the dynamic schemes support efficient multi-keyword ranked search.

This project proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. In order to obtain high search efficiency, we construct a tree-based index Multikeyword Rank Search Scheme For Unindexed Encrypted Cloud Data structure and propose a ―Greedy Depth-first Search (GDFS)‖ algorithm based on this index tree. Due to the special structure of tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents. The secure KNN algorithm is utilize to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. . In order to calculate the TF value of the search keyword we use a pattern matching algorithm like Naïve algorithm which indicates the occurrence of that particular keyword. In existing system, The techniques of data updating are utilizes effectively but there is big problem in working with sharing keys and decrypted data with other users which may disturb the security as well in this a unencrypted index key is used for ranking which may break security as well. So that we proposed a mechanism in which the encrypted index term key will get generated and perform the evaluation for the multi keyword searching in all encrypted cloud storage.

This project helps to implement metadata based keyword search. Metadata based keyword search means search engine that powers a portal search based on a specific metadata schema. It encourages to implement usability based ranking optimization i.e. to check how many times a particular file has been accessed .It supports privacy preserving over shared data to cloud by means of encrypting data. As well to implement efficient ranking system based on term frequency generation. The TF-IDF module is used for page ranking. We begin our work with the architecture of the proposed system.

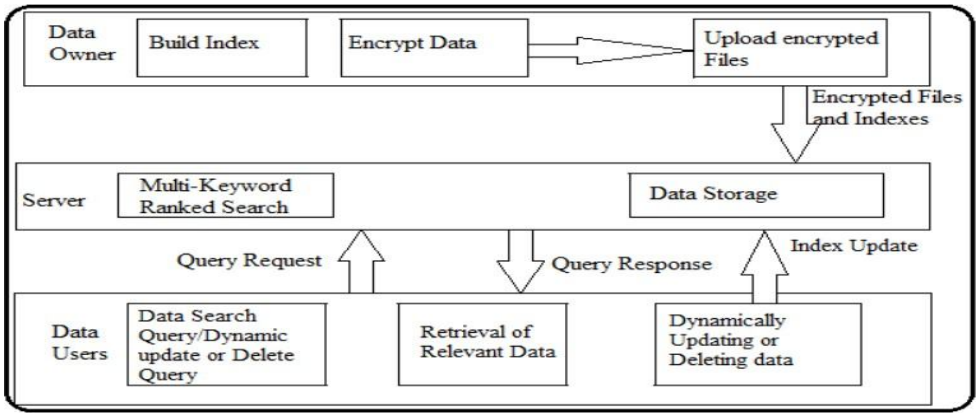


Fig 1 Architecture of the proposed system

The architecture in our proposed system involves three different entities: the data owner, the data user, and the cloud server as shown in Fig.1 The data owner has the document collection(files) which will be outsourced into the cloud. The data owner encrypts all files in the form of suppose C before outsourcing it to the cloud in order to protect the sensitive data from unauthorized entities. Then it outsources the encrypted data files with encrypted index to the cloud server also distributes secret keys & information about the tree construction to the desired data users. Data users have secret keys to access the documents .. While searching the system will generate an encrypted search query based on the keyword entered by authorized user. Given the encrypted search query & a parameter k , the cloud server will search the index I and then return top-k most relevant documents to the user based on the concept of page ranking. The result of search is a set of encrypted documents containing the entered keywords & in our proposed work It is well ranked. As the data owner outsources the encrypted data files into cloud server via Cloud service provider , there remains no control of him over it. This raises privacy issues in the cloud. Though CSP provides some standard security mechanism to protect the data from attackers still it is hacking. Therefore we need an efficient and secure mechanism to protect the privacy of sensitive outsourced data in the cloud.

II. LITERATURE REVIEW

Zhihua Xia, Xinhui Wang, Xingming Sun and Qian Wang[1] presented 'A secure and Dynamic Multikeyword Ranked Search scheme over Encrypted cloud data' in 2016. Their work supports not only the accurate multikeyword ranked search but also the dynamic deletion and updation of documents. A special keyword balanced binary tree is constructed to achieve better efficiency than linear search with the help of Greedy Depth-first search algorithm.Parallel search process is carried out to further reduce the time cost. Zhangji FU Xingming Sun QiLiu LuZHOU Jiangang SHU[2] have presented ' Multi-Keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing' in 2015 .They proposed a practical ,efficient and flexible searchable encryption scheme which supports both Multikeyword ranked search and parallel search. To improve search efficiency, they design a tree-based index structure which supports parallel search to take advantage of the powerful computing capacity and resources of the cloud server.Searchable Symmetric Encryption (SSE) has been proposed by Chang Liu, Liehuang Zhu, Jinjun Chen [3] under the name 'Efficient Searchable Symmetric Encryption for Storing Multiple Source Data on Cloud' in 2017. They propose a notion of Multi-data –source SSE which allows each data source to build a local index individualy and enables the storage provider to merge all local indexes into a global index afterwards. Tianyue Peng ,Yaping Lin,Xin Yao,Wei Zhang [4] ' in 2018 proposed 'An efficient ranked multi-keyword search for multiple data owners over encrypted cloud data'. Their work is especially for multi-owner scenario. They proposed a tree-based ranked multikeyword search scheme for multiple data owners. Considering large amount of data in the cloud , they utilize the TF*IDF model to develop a multi keyword search and return the top-k ranked search results. Tianyue Peng ,Yaping Lin,Xin Yao,Wei Zhang [4] ' in 2018 proposed 'An efficient ranked multi-keyword search for multiple data owners over encrypted cloud data'.

Their work is especially for multi-owner scenario. They proposed a tree-based ranked multikeyword search scheme for multiple data owners. Considering large amount of data in the cloud , they utilize the TF*IDF model to develop a multi keyword search and return the top-k ranked search results. N Cao,C Wang, M Li, K.Ren and W .Lou [6] have proposed 'Privacy preserving multikeyword rank search over encrypted cloud data' in 2011. They attempt to define and solve the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). C.Orencik, M.Kantareio and E.Savas [7] proposes 'A practical and secure multikeyword search method over encrypted cloud data' .They proposed an efficient privacy-preserving search method over encrypted cloud data that utilizes minhash function. Author Miss Deepali D.Rane & Dr. V. R. Ghorpade [8] have proposed a work in which two of the privacy preserving issues about accessing the cloud data has been identified i.e. acuteness of keywords sent in queries and the data fetched as a result of those queries. Both of them shall be hidden. To keep the privacy of documents, it should get encrypted before outsourcing to the cloud. Privacy of the documents has been achieved using symmetric key cryptography algorithm i.e. Twofish.

III.PROPOSED SYSTEM ANALYSIS

In existing system the challenge is symmetric searchable schemes. It requires huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical thing. Existing System methods are not practical due to their high computational overhead for both the cloud sever and user. In the proposed scheme, the data owner is responsible for updating information and sending them to the 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, meanwhile reserving the ability to support multi-keyword ranked search. In addition, as the most of works about searchable encryption, our scheme mainly considers the challenge from the cloud server. To overcome the issues of existing system we attempt a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. In order to obtain high search efficiency, we construct a tree-based index structure and propose a ―Greedy Depth-first Search (GDFS)‖ algorithm based on this index tree. Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the updating and insertion of documents. The secure KNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. Our contributions are summarized as follows:

We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection. Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing ―Greedy Depth-first Search‖ algorithm and KNN algorithm. The objectives of the proposed system can be summarized as follows:

1) We designed a searchable encryption scheme that supports both the accurate multi keyword ranked search and flexible dynamic operations like insertion and updating.
2) Due to tree-based indexed structure the search complexity of the proposed is kept to logarithmic.
3) The proposed scheme can achieve higher search efficiency by executing ―Greedy Depth-search ‖ algorithm.
4) The objective is to design a model which will provide not only encrypted data files but encrypted index file too.
5) To implement metadata based keyword search.
6) To implement usability based ranking optimization. It means to check how many times a particular file has been accessed.
7) To implement privacy preserving over shared data to cloud by means of encrypting data.
8) To implement efficient ranking system based on term frequency generation. TF-IDF model is used for page ranking.

IV.REQUIREMENT ANALYSIS

A. Software Tools

Java (jdk1.7, jdk 8.0) as a working environment.
Front-end : Eclipse Luna software by Java EE.
Server : Tomcat 7.0 server
Database : My SQL 5.5 database
Operating System : window 7,window 8,window 10

.

B. Work Flow of the proposed System

1) Step 1: Authentication Process

a) Registration of new user.
b) Admin first confirm the pending user.
c) Allot cloud to confirmed user.
d) Now authorized user can login with ID and PWD .
e) Admin log out.

2) Step 2: Uploading File

a) Authorized Data user now log in to the system .
b) Data owner i.e. authorized user now can outsource multiple data files to cloud .
c) Internally system will encrypt the data files and index , to outsource it.
d) Data files and index will be encrypted before outsourcing.

3) Step 3: File Retrieval

a) Multiple Users can search the data uploaded on server by multiple keywords.
b) Required data can be downloaded to the local machine.

The detail description about the working of the system can be summarized into two phases: The initial phase, The retrieval phase.

C. Initial Phase

1) Key Generation: Key generation is the process of generating keys for cryptography. The key is used to encrypt and decrypt the data at both end. Symmetric key algorithm like AES & DES is used .Symmetric key algorithm uses a single shared key keeping data secret . Public key algorithm uses the public key and private key. Data owner encrypt the data with the public key, only the holder of the key can decrypt this data.

2) Index Generation: Index can be created for a table to improve the performance of queries issued against the corresponding table. In our proposed system we are having Application based Index generation depending on queue structure. All primary key columns are in the primary index of the table automatically.

3) Privacy Preserving [AES]: After index creation to ensure the privacy of index & files the data owner encrypts both index and file collection. Data owner encrypts index and files using AES .

Pseudo code for AES Algorithm:

```
Void Cipher (byte []in ,byte[]out,byte[]w)
{ byte[][]state=new byte[4][Nb];
 State =in;
 AddRoundKey(state,w,,0,Nb-1)
 for( int round=1;round <Nr; round++)
{ SubBytes (state);
Shift Rows (state);
MixColumns(state);
AddRoundKey(state,w,Round*Nb,(round+1)*Nb-1)
}
 SubBytes(state)
 Shift Rows(state);
 AddRoundKey(state,w,Nr*Nb,(Nr+1)*Nb-1)
Out=state;
 }
```

D. Retrieval Phase

The second phase i.e. retrieval phase , authorized user retrieves the files from the CSP through ranked keyword search. This phase consist of:

1) Ranked Keyword Search: In this operation ,the cloud server searches for the matching files after receiving the search request. The cloud server first finds the matching entries using pattern matching algorithm, if server gets matching file as per their weitage then the server ranks the matched files as per their weitage and sends top k-most relevant files Ci= {c1,c2,c3…..ck }to the user.

a) Step 1: For each level I from 1 to n do
b) Step 2=T(wi)…T(wi) is the term frequency
c) Step 3: Rank (Ci) , highest level that match with query
d) Step 4: end of step 2
e) Step 5: end of step 1
f) Step 6: send C={c1,c2……ck} to the user.

To search the files, we use tree-based index structure to get the corresponding file list. For each file, each tree-level stores an index for frequent keywords of that file . .If matching file found as a result of the comparison in the first level then this process continues to the next level. Thus the overall search time is almost efficient as on unencrypted data. Our search focuses on top-k retrieval.

2) *Data Decryption[DES]:* After receiving the matched files from CSP for corresponding search request, the authorized user decrypts them with the private key given by DO and obtain their plain text. DES is a block cipher, and encrypts data in blocks of size of 64 bit each, means 64 bits of plain text goes as the input to DES, which produces 64 bits of cipher text. The same algorithm and key are used for encryption and decryption.

Pseudocode of decryption:

```
Void InvCipher(byte[]in,byte[]out,byte[]w)
{ byte[] state=new byte[4][Nb];
state=in;
AddRoundKey(state,w,Nr*Nb,(Nr+1)*Nb-1)
for(int Round=Nr-1;round>=1;round--)
{ InvShiftRows(state);
InvSubBytes(state);
AddRoundKey(state,w,round*Nb,(round+1)*Nb-1)
InvMixColumns(state);
}
InvShiftRows(state);
InvSubBytes(state);
AddRoundKey(state,w,0,Nb-1);
Out=state;
}
```

V. DETAIL DESIGN

A. Greedy DFS Algorithm

A recursive procedure upon a tree i.e.Greedy DFS algorithm is used in search process of unencrypted multikeyword ranked search scheme. Let‘s construct a list where the result are stored denoted as Rlist. Rscore is the relevance value or weitage of the document F. The relevance score or weitage is calculated by product of index vector in the tree node and the query vector. Can be denoted as,

$$R(F)=Iu*Q \quad \text{……Equation 1}$$

During the search process number of accessed documents, lets say _k‘ stored in Rlist with largest relevance value to the query. Elements in the list are ranked in descending order as per their weitage & updated timely during search process. While constructing the tree index , we first generate leaf nodes from the documents . Then the internal tree nodes are generated based on the leaf nodes. Thus during the search process only one part can be accessed at a time.

Pseudocode:

1) if node u is not a leaf node then
2) if Rvalue(Iu,Q) > Kth value then
3) GDFS(u.hchild);
4) GDFS(u.lchild);
5) else
6) return
7) end if
8) else
9) if Rvalue(Iu,Q)> kth value then
10) delete the element with smallest relevance value from Rlist
11) Insert a new element {Rvalue(Iu,Q),u} & sort all the element of Rlist

12) Endif
13) Return
14) endif

B. KNN Algorithm

KNN works by finding the distances between a query vector Q and all the examples in the data , selecting the specified number examples (K) closest to the query Q, then votes for the most frequent label. Many analyses have found that the kNN algorithm manages a very excellent performance in their analysis on distinct data sets.

The steps in the Algorithm is as follows…

Step 1: Give the query keyword.
Step 2: Select all data related to that query word.
Step 3: If no results found.
Stop
Step 4: else
Retrieve all the post belongs to the keyword and in given range.
Step 5: Take K value and range If (k > no. of posts)
Repeat step 4 with increasing range
Step 6: Calculate distance between all posts and user‘s query
Step 7: Sort according to the distance (ascending order)
Step 8: else
Take first K results.
Send result.
Step 9:Exit

C. Pattern Matching Algorithm

Tf-idf stands for *Term frequency-inverse document frequency*. The tf-idf weight is a weight often used in information retrieval and text mining. Variations of the tf-idf weighting scheme are often used by search engines in scoring and ranking a document‘s relevance given a query. This weight is a statistical measure used to evaluate how important a word is to a document in a collection . The importance increases proportionally to the number of times a word appears in the document.

TF is the Normalized term frequency that indicates the number of occurrencess of a particular term *t* in document *d*. Therefore,

$$Tf(t,d)=N(t,d) \quad . \quad \text{……Equation 2}$$

Tf(t,d)=term frequency for a term t in document d

Let‘s consider a simple pattern matching algorithm called Brute Force.

Algorithm :BruteForceStringMatch

1) Searching for a pattern, P[0...m-1], in text, T[0...n-1]
2) for i ← 0 to n-m do
3) j ← 0
4) while j < m and P[j] = T[i+j] do
5) j++
6) if j = m then return i
7) return -1

D. Keyword Binary Index Tree Structure With Index Building Algorithm

The balanced binary tree is used widely in optimization problems.[1] It‘s a dynamic data structure where node stores a vector D. The elements of vector D are nothing but normalized Tf values.

Steps to construct a Binary Index tree:

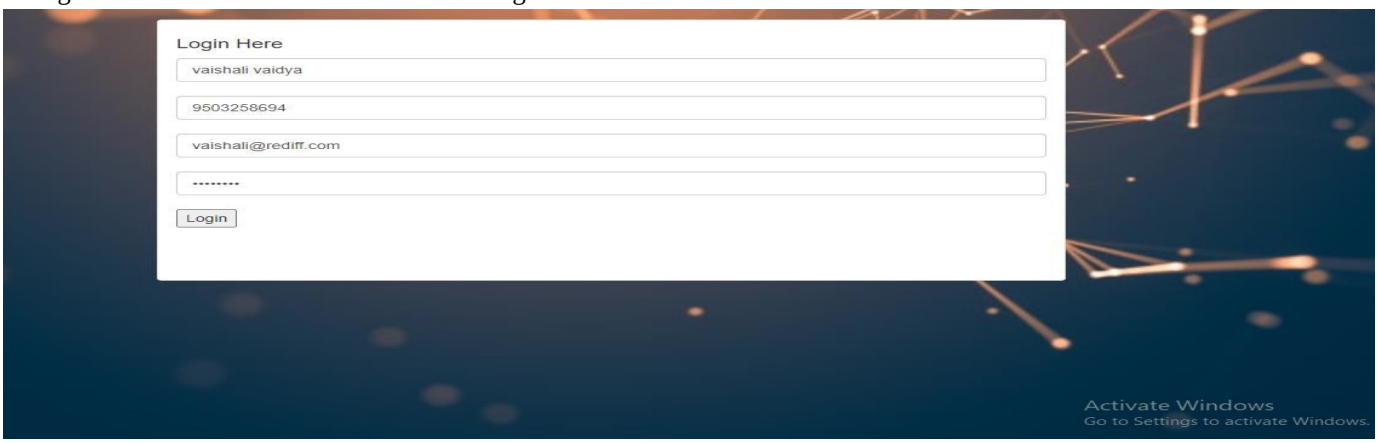
1) Suppose root of the tree is dummy.
2) We can represent binary index tree as an array. We start inserting elements from first index.
3) Each node of the tree has an index and value.

4) To construct a tree we first initiate all node with 1.
5) Suppose BT[] is an array . We represent node‘s index in term of sum of power of 2.
6) Now based on this representation we store the value into that node.

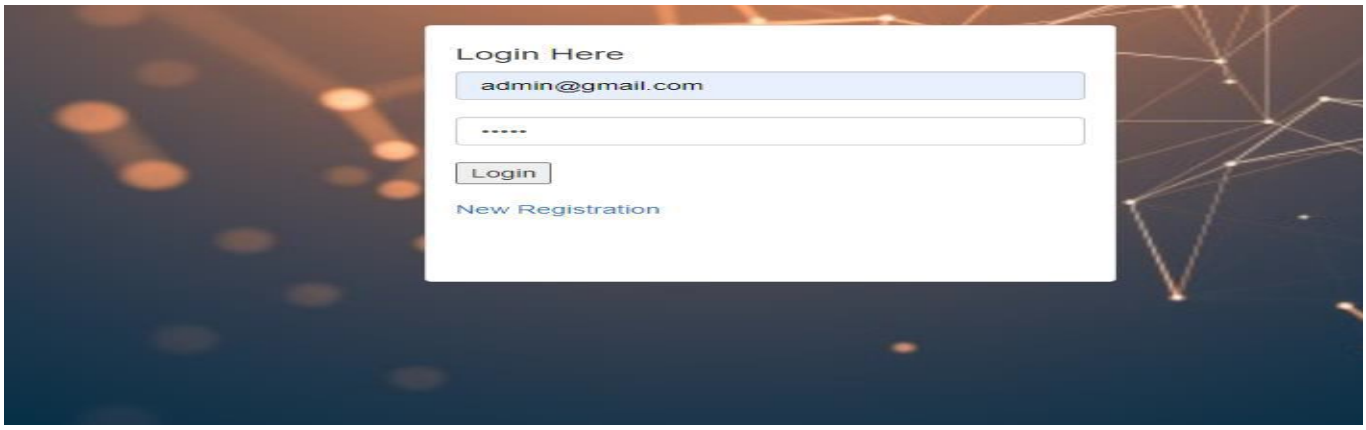
VI. SYSTEM IMPLEMENTATION

This section of work includes Screen shots of entire work from both sides i.e Admin part & data owner and data user part.

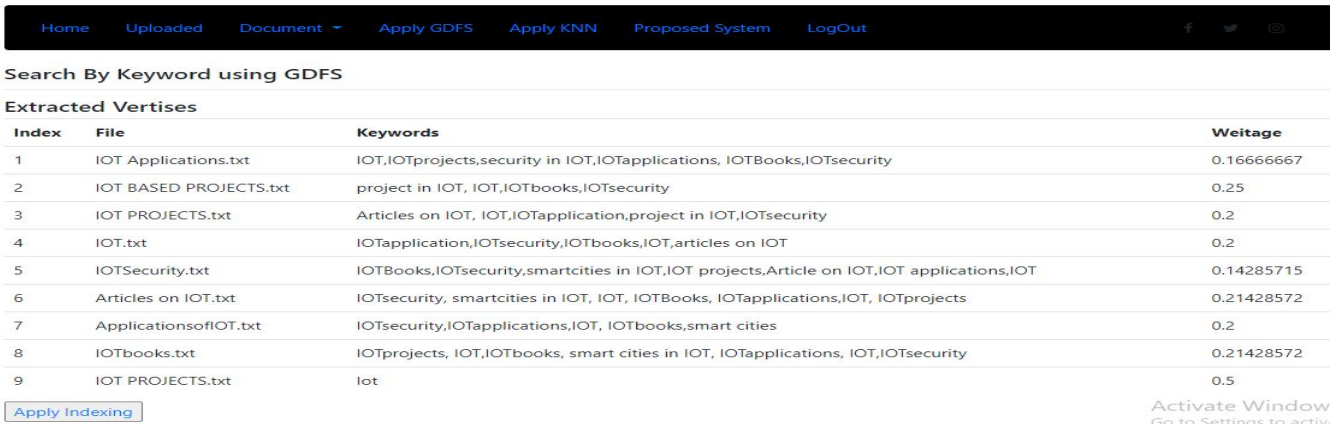
1) Registration of the user- A new user has to register first with his/her information.



2) Admin Login—Admin login to the system with ID-admin@gmail.com and password- admin



3) Apply GDFS-- Next operation is performing greedy DFS. This search operation is performed on the plaintext.

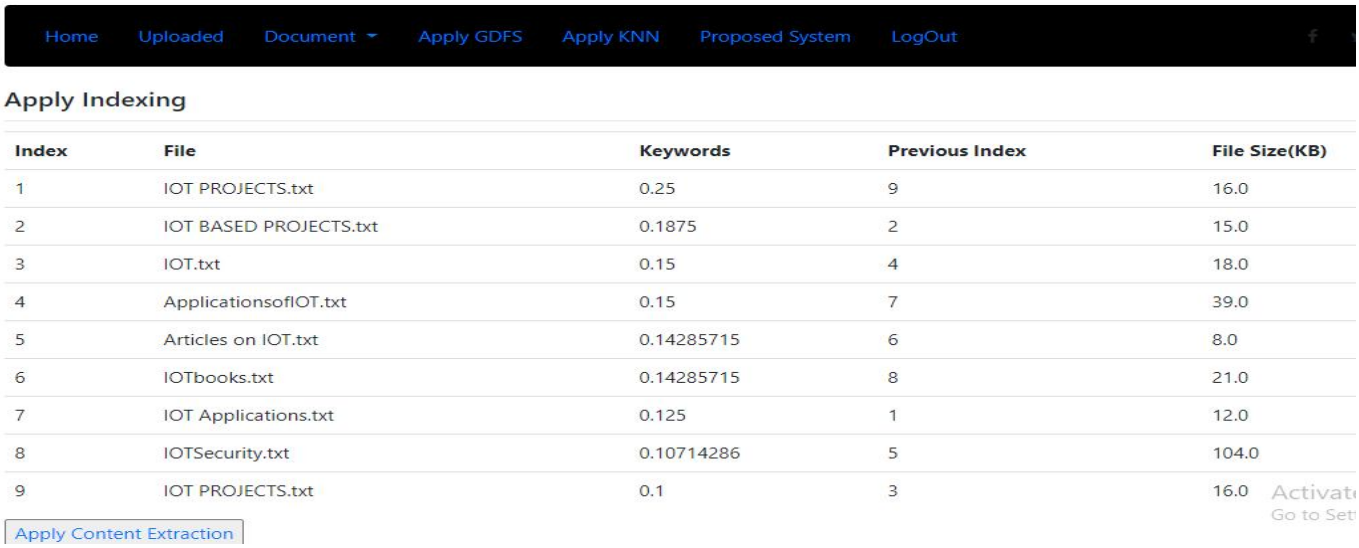


4) Apply KNN-- Next operation is to perform KNN algorithm on the encrypted data.

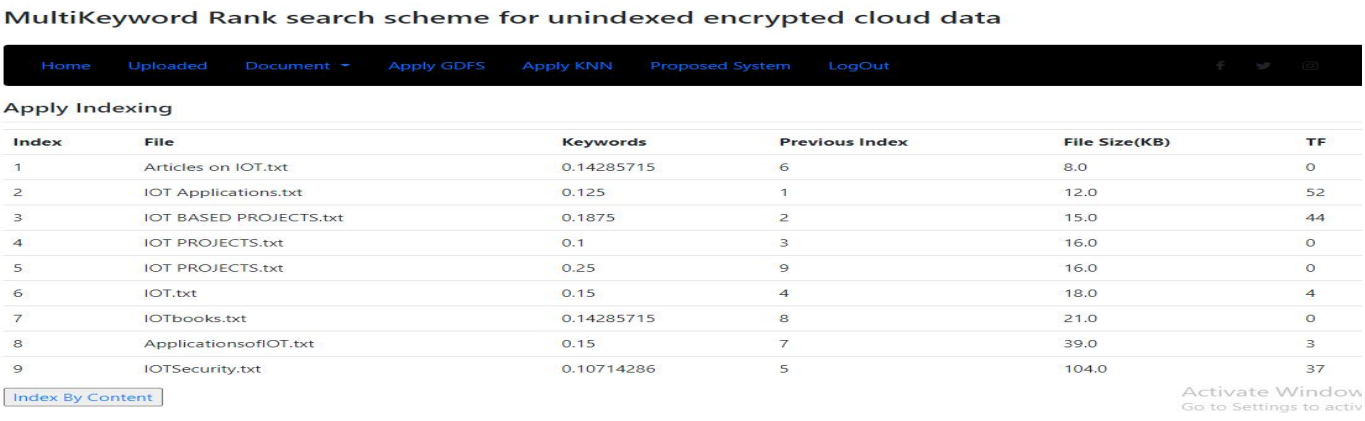
Extracted Vertises

Index	File	Keywords	Distance
1	IOT Applications.txt	*e?" O9v???F@????□8??□??J?W??}?8?]?i□?□+??=?zI.???yD??? ?□2V??\?□□???%?,?t??*?□?□ ?Ø"?3??VKk$?>□-gR??□/?? G?🡅??? M?y9??.Y&?.*?m□C??□?'□/???OD7□?N?7Q??□□AE□&?□?&??5□???|_??,?&????S2□[??.k??□?:?~z????n?g?H?□B?? +?^?□□??□?'?T[??;?t;6v0□?x7?=????"n}?	6
2	IOT BASED PROJECTS.txt	`?v1z ??1?h`???kp????]iy?/????y????i?Wi)?6z?□%???7□>co?????□?|\???Z?1@?? :?□?9e?^?Z??C]D?qzh□?X?s\□????t?>?□?? t□?□?240???`,{???????#F?????,□?L??I□?N_H? V?o?□???□?yâ?iq??4I??🡅?_k??tV?kQ?□??????-1???zi:□??9?□u??.□?]*□uX?;??? i?+ ???□????N??Vq□?□???	4
3	IOT PROJECTS.txt	6□?&? ? ????□????`?□]???)E|??C?2#J3 ????□?□?p□?&?□3?E?%c□1M□?C?□□3w?@;4?)???Z?#???jt□?%??□?u?????S-?-???? m(??□□ ??c[_??□E??a?1?mw?D G[=+??1W?:]=?N??kE?I□□□ ?□□???□Dk🡅A????□Y?wA?rP$?? 9R?/?□y??□?3|???t ?? |????□?I???= :?;?□C?+r??□B□??I??p?K??]+□??	5
4	IOT.txt	m□□?□?C??6>Z'?????&>□Z?f??9n??W?C?>???>S??□,?(???r?t? ????w_?u?zu.□,X???%??FS??a?□????H=?=>??&A?£?D? #?/?□????□?.??C??6?/;`?q?`s:??"/?n?□IW=i??]x??O ??G?o?c?u?u?_??□I??□?□?)O?a□???????□6?□RF????n□#3Pj_?@X□?:-?0??y?~X??????Bz?u???g??gd??G??E?	2
5	IOTSecurity.txt	6?9	2
6	Articles on IOT.txt	????h??`/K ?V?? ?wg??0X□O0J□[?4???Y????,??□?g8??m?D?5□??`**C???c????5=????e\?o?$??"W□p□?□???? 7?Ay?□}?)E?? O+`~$□??s??2??/?]?a|r`?h 9J?Bb□K-K??-???M*□?T9???□□□p?w???T??~??y?#□?Lt□?wy?2t???`??)?3??g?□?`???8D□?{	1
7	ApplicationsofIOT.txt	j??□jy□???F4?BC??Q?????H??N7	1
8	IOTbooks.txt	@??□??□???&?□?□?DB??pl□? ??w□s???r????q?1??b□□?s??????□?$?1???8????□?□????bh□?J?G)?4□3?@???6□□?'□?]j?	7
9	IOT PROJECTS.txt	~8:?'T□o?P???□? ????ZP□??D?6[?□tcL{F□LN8O?)□?????1`?□[??!?-?C~???D?□??i5?□□□a??□H?]???fW&W???□??□?fj?W??? N?□□□?S□L???>F??p??e??□?d□□?□+?5,=?□y???'??8??□???!?|?|?3"??o???□'??*? %??E?q?d?-?□ □?`??□c???yy???P???□?!?? 0?? ?□ T??@?L???L7?????Q?□o??T???	2

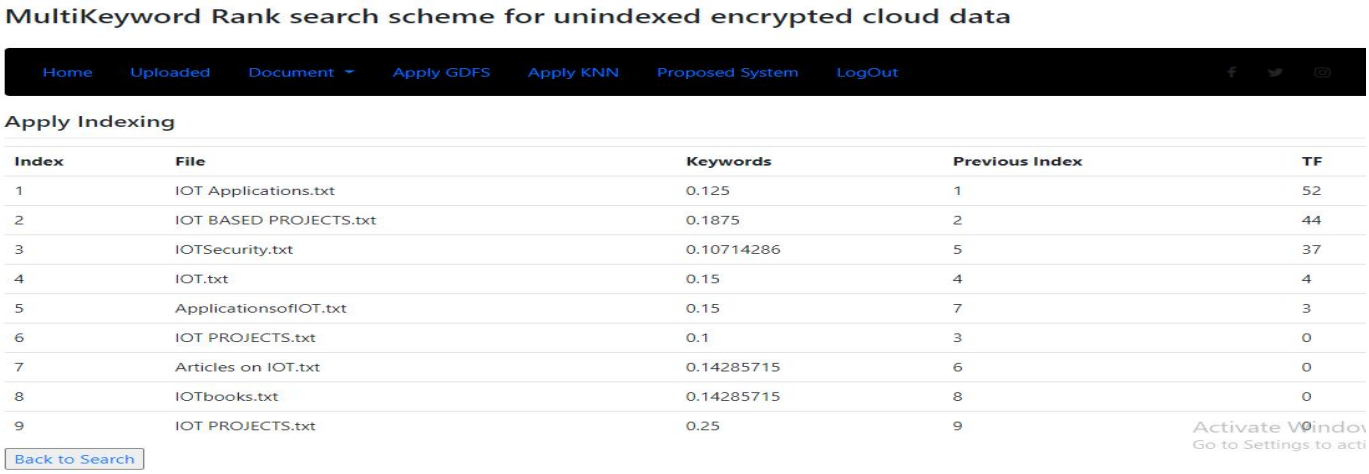
5) Extraction of metadata by document size



6) Index ranking by file size with effective generation of TF value



7) Index formation by content with effective generation of TF value



VII. RESULT ANALYSIS

Result analysis is drawn after the execution of system. Input is given in the form of single keyword, double keyword and multi keywords. All the inputs are checked for GDFS algorithm , KNN algorithm and proposed system.

Table 1: Determine weitage of Edges.Weitage or the relevance value is calculated by means of using equation 1.

Index	File	Keywords	Weightage
1	IOT Applications.txt	IOT,IOTprojects,security in IOT,IOTapplications, IOTBooks,IOTsecurity	0.16666667
2	IOT BASED PROJECTS.txt	project in IOT, IOT,IOTbooks,IOTsecurity	0.25
3	IOT PROJECTS.txt	Articles on IOT, IOT,IOTapplication,project in IOT,IOTsecurity	0.2
4	IOT.txt	IOTapplication,IOTsecurity,IOTbooks,IOT,articles on IOT	0.2

Table 2: Applying Indexing

Index	File	Keywords	Previous Index
1	IOT PROJECTS.txt	0.5	9
2	IOT BASED PROJECTS.txt	0.25	2
3	Articles on IOT.txt	0.21428512	6
4	IOTbooks.txt	0.21228572	8

Table 3:Calculate distance using KNN algorithm for multi keyword rank search

Distance is calculated by comparing the query vectors and other data keywords. For example query vectors IOT and IOTsecurity is compared with other keywords in metadata. And Output is displayed with its position in the metadata.

Index	File	Keywords	Distance
1	IOT Applications.txt	*e?" O9v???F@????8????J?W??}?8?]?i?+??=?z!.???yD??? ?2V??\????%?, ?t??*?? ?Ø"?3??VKk$?>-gR??/??G???? M?y9??.Y&?.*?mC???'/???OD7	6
2	IOT BASED PROJECTS.txt	`?v1z ??1?h`???kp????]iy?/????y????i?Wi)?6z?-%???7>co?????\???Z?1@?? :? ?9e?^?Z??C]D?qzh?X?s\	4
3	IOT PROJECTS.txt	6?&? ? ????????`?]???)E\??C?2#J3 ?????-?p?&?3?E?%c1M?C?3w?@ ;4?)???Z?#???jt?%???u?????S-?-????m(?7???c[_??E??a?1?mw?D	5

Table 4: Sort Indices using KNN algorithm

Index	File	Keywords	Previous Index
1	Articles on IOT.txt	IOTsecurity, smartcities in IOT, IOT, IOTBooks, IOTapplications,IOT, IOTprojects	6
2	ApplicationsofIOT.txt	IOTsecurity,IOTapplications,IOT, IOTbooks,smart cities	7
3	IOT PROJECTS.txt	Iot	9

Table 5:Select Top-K results

Index	File	Keywords	Previous Index
1	Articles on IOT.txt	IOTsecurity, smartcities in IOT, IOT, IOTBooks, IOTapplications,IOT, IOTprojects	6
2	ApplicationsofIOT.txt	IOTsecurity,IOTapplications,IOT, IOTbooks,smart cities	7

Table 6: Search by keyword using GDFS for proposed system

IOTsecurity applications IOTbooks IOT

Keyword weitage

Index	File	Keywords	Weightage
1	IOT Applications.txt	IOT,IOTprojects,security in IOT,IOTapplications, IOTBooks,IOTsecurity	0.125
2	IOT BASED PROJECTS.txt	project in IOT, IOT,IOTbooks,IOTsecurity	0.1875
3	IOT PROJECTS.txt	Articles on IOT, IOT,IOTapplication,project in IOT,IOTsecurity	0.1
4	IOT.txt	IOTapplication,IOTsecurity,IOTbooks,IOT,articles on IOT	0.15

Table 7: Extraction of Metadata

Index	File	Keywords	Previous Index	File size(KB)
1	IOT PROJECTS.txt	0.25	9	16.0
2	IOT BASED PROJECTS.txt	0.1875	2	15.0
3	IOT.txt	0.15	4	18.0
4	ApplicationsofIOT.txt	0.15	7	39.0

Table 8: Apply content Extraction

Index	File	Keywords	Prev. Index	File size(KB)	TF value
1	IOT PROJECTS.txt	0.25	9	16.0	0
2	IOT BASED PROJECTS.txt	0.1875	2	15.0	44
3	IOT.txt	0.15	4	18.0	4
4	ApplicationsofIOT.txt	0.15	7	39.0	3

Table 9: Index by Document size

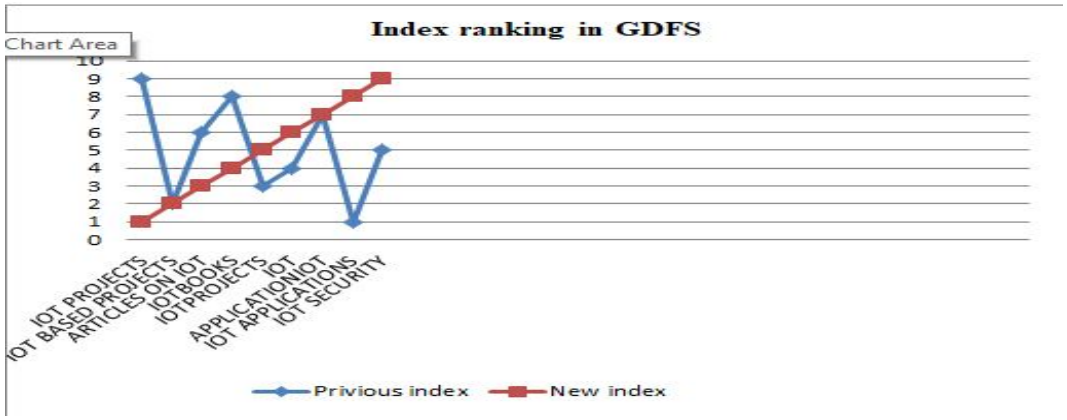
Index	File	Keywords	Prev. Index	File size(KB)	TF
1	Articles on IOT.txt	0.14285715	6	8.0	0
2	IOT Applications.txt	0.125	1	12.0	52
3	IOT BASED PROJECTS.txt	0.1875	2	15.0	44
4	IOT PROJECTS.txt	0.1	3	16.0	0

Table 10: Index by content wise.

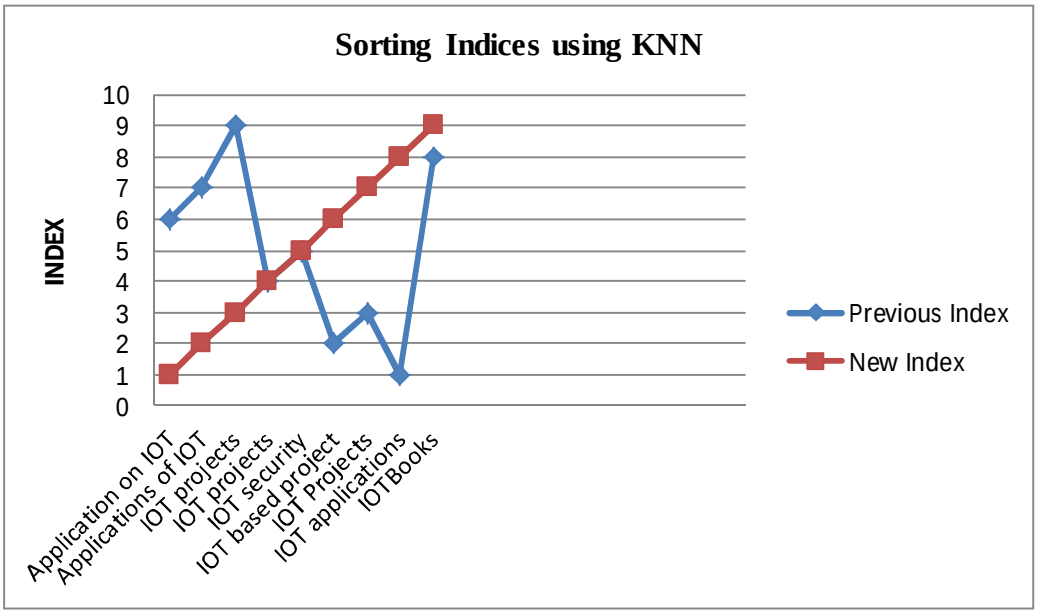
Index	File	Keywords	Prev. Index	TF
1	IOT Applications.txt	0.125	1	52
2	IOT BASED PROJECTS.txt	0.1875	2	44
3	IOTSecurity.txt	0.10714286	5	37
4	IOT.txt	0.15	4	4

A. Graphical Representation of Results Obtained

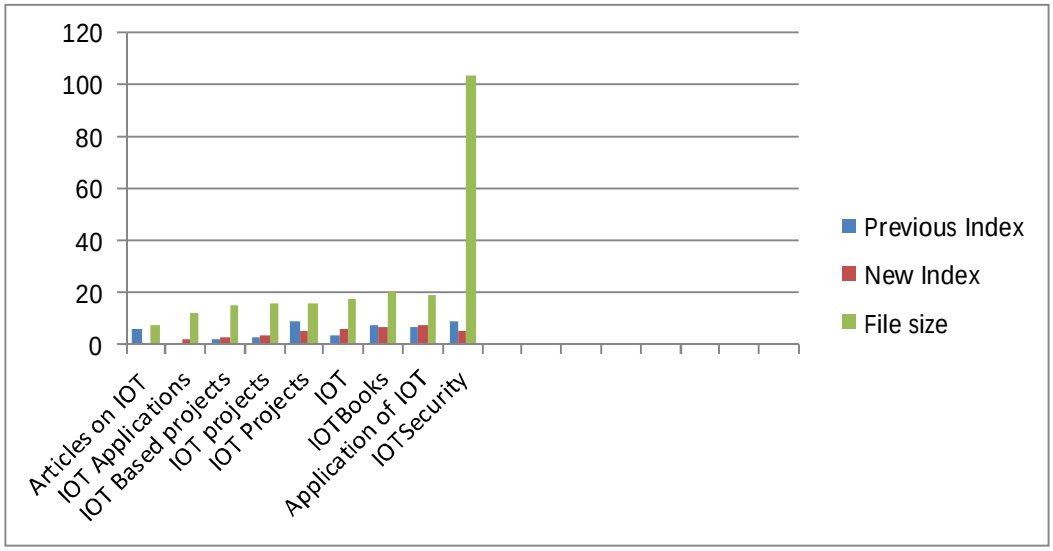
1) Index Ranking graph for Multi keyword rank search using GDFS



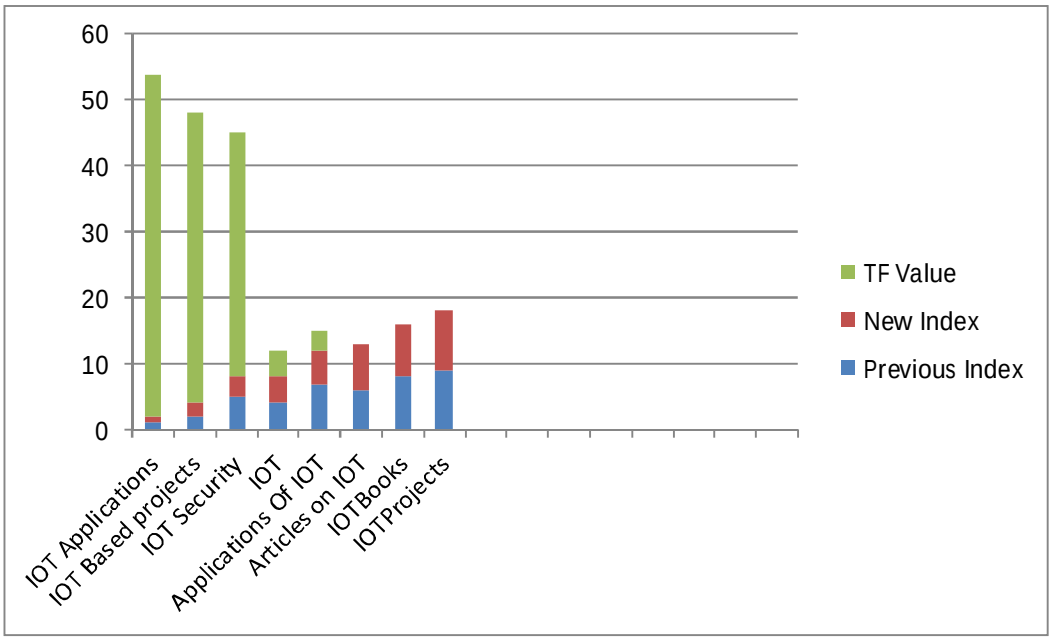
2) Sorting Indices graph for multi keyword rank search using KNN algorithm



3) Graphical representation of Index ranking By document size in proposed system analysis



4) Graphical representation of Index ranking by content in proposed system analysis



VIII. CONCLUSION AND FUTURE SCOPE

In this study, a secure multi keyword ranked search is proposed which not only supports the accurate keyword search for single as well multi keyword ranked search but also the dynamic operations like insertion and updating of documents. To obtain better efficiency we construct a balanced binary tree as index and perform search operation using Geedy DFS algorithm. Using a secure KNN algorithm the search efficiency of the proposed system is enhanced. A metadata based keyword search is implied successfully by performing rigorous operation with the help of single and multi keywords. TF value is calculated accurately to know the occurrence of particular keyword in a file. Indexing is done successfully in each experiment on the basis of weitage value, distance of query keyword as compare to other keywords, and on the basis of file size and content wise.

There are still many challenges in the symmetric SE schemes. Actually in the dynamic operations like insertion and updating, data owner needs to store unencrypted index tree , such kind of work is not suitable for cloud environment.

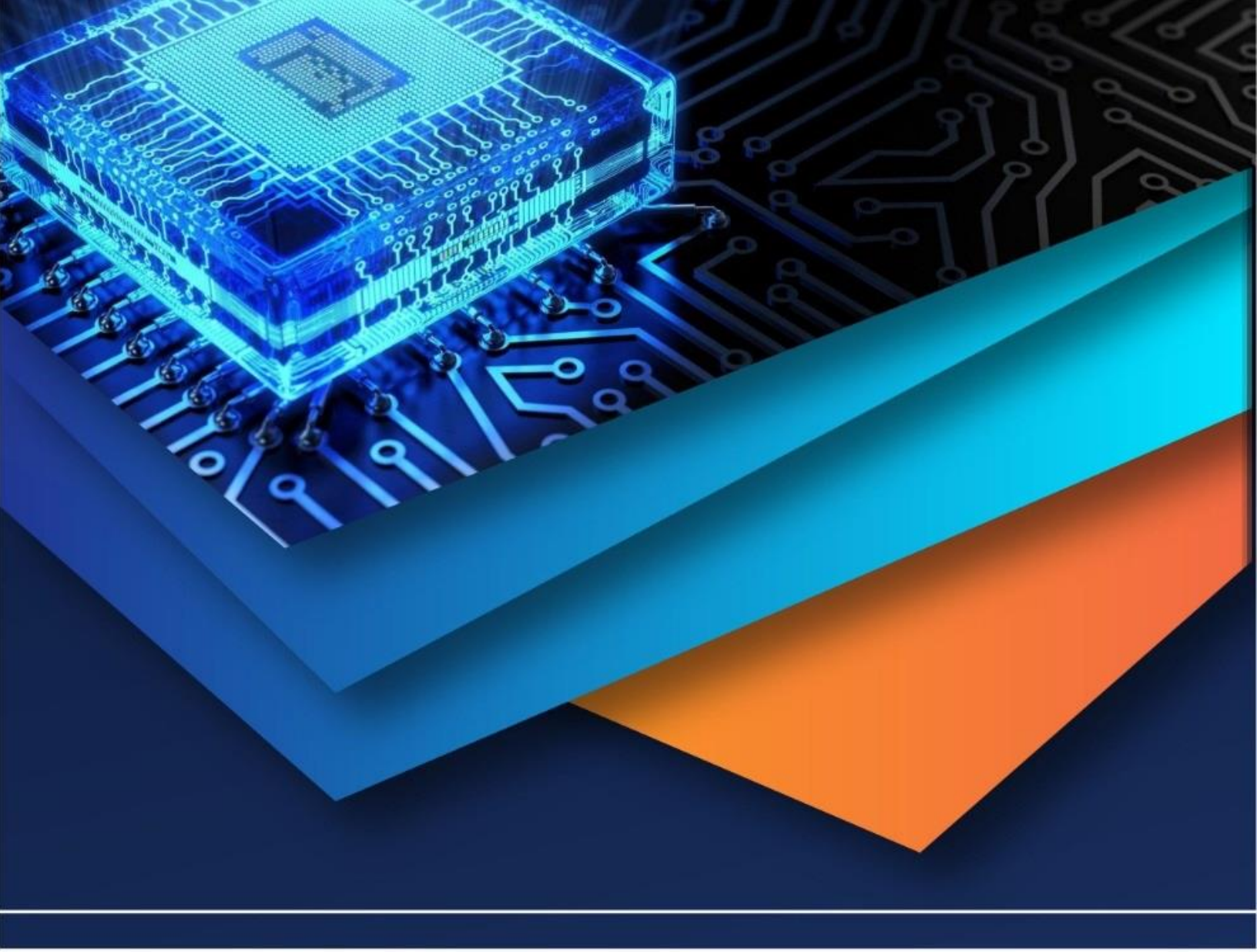
It needs to be work out by data owner. As the proposed scheme is multi-user ,all the users share the same secure key. So revocation of user is a big problem . We need to rebuild the index and distribute the new secure keys to all authorized users.
One more problem is of security like internal attacks as well external . Query and Index confidentiality need to be maintained. While decrypting the data from the cloud by user shall not receive corrupt. Data. So security is a big concern in such models.
In such schemes it is also possible that not every data user is honest. Such dishonest data user may perform search and distribute decrypted document to unauthorized ones. Or they may share their secret keys .
These kind of problems in case of searchable encryption models and cloud computing needs to be improve in future.

IX. ACKNOWLEDGEMENT

I feel immense pleasure to express deep sense of gratitude and indebtedness to my guide Dr A.A.Bardekar, for constant encouragement and noble guidance. I express my sincere thanks to Dr.V. K. Shandilya, Head of Department, Computer Science & Engineering, and the other staff members of the department for their kind co-operation.
I express my sincere thanks to Dr. S.M.Kherde Principal, Sipna College of Engineering & Technology for his valuable guidance. I also express my sincere thanks to the library staff members of the college.Last but not the least we are thankful to our friends and our parents whose best wishes are always with us.

REFERANCES

[1] Zhihua Xia, Xinhui Wang, Xingming Sun and Qian Wang," A Secure and dynamic multikeyword ranked search scheme over encrypted cloud data"IEEE transactions on parallel and distributed systems,vol 27, no 2 february 2016.
[2] Zhangji FU Xingming Sun QiLiu LuZHOU Jiangang SHU, "Multi-Keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing' IEICE TRANS.COMMUN.VOL.E-98-B,NO.1 JANUARY 2015
[3] Chang Liu, Liehuang Zhu, Jinjun Chen,"Efficient Searchable Symmetric Encryption for Storing Multiple Source Data on Cloud" 978-1-4673-7952-6/15 $31.00 © 2015 IEEE
[4] Tianyue Peng ,Yaping Lin,Xin Yao,Wei Zhang, "An efficient ranked multi-keyword search for multiple data owners over encrypted cloud data" JOURNAL OF LATEX CLASS FILES, VOL. 6, NO. 1, JANUARY 2015
[5] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Secur. Privacy, 2000, pp. 44–55
[6] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in Proc. IEEE INFOCOM, Apr. 2011, pp. 829–837.
[7] C. Orencik, M. Kantarcioglu, and E. Savas, "A practical and secure multi-keyword search method over encrypted cloud data," in Proc. IEEE 6th Int. Conf. Cloud Computing., 2013, pp. 390–397.
[8] Deepali D.Rane & Dr.V.R.Ghorpade ," Multi-User Multi-Keyword Privacy Preserving Ranked Based Search Over Encrypted Cloud Data" International Conference on Computing (ICPC)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY