A Dynamic and Secure Multiple-Keyword Ranked Parallel Search

Jayanth J[1] Mrs Jayasheela C.S[2]
[1]M.Tech Student [2]Assitant Professor
[1,2]Department of Information Science Engineering
[1,2]Bangalore Institute of Technology, Bangalore, India

Abstract— Being progressive increment in the utilization of Cloud computing innovation by more users, the approved owners of information are intrigued to send their heterogeneous data to cloud with higher security and a large amount of data need to be easily searched and retrieved from the cloud storage management.. Since there is higher odds of burglary of such data, clients of the cloud are more worried about keeping up the security in the trading and searching of such information. Keeping in mind the end goal to get the ease of searching documents and higher rate of security, clients of cloud innovation inspired to include searching methods to retrieve the documents and encryption methods for encoding such information respectively. In this paper for faster and secure searching of the data in the cloud storage is satisfied by using the Parallel Search concept in the existing multiple keyword ranking based search approach. This concept in the paper allows the reduction in time cost to search the data in the cloud since Parallel Search concept involves multithreading schema in which the large amount of search requests can be divided into multiple batches and handled separately by the different threads simultaneously. Also the project handles the secrecy of the keyword information by encrypting it using the MD5 hashing method.

Key words: Parallel search architecture for searching documents, MD5 hashing, Multithreading, Database, Dynamic searchable encryption

I. Introduction

At present, Cloud data storage have become the most well liked solution to provide the fast, well organized and accurate access of huge data transferred through wireless and wired media. Globally all people have started using this technology for exchange of data like pictures, documents and videos etc with others. Most importantly all the organizations, institutions and enterprises use this technology for secure exchange of information and data backups in order to prevent data theft by the third person and loss of valuable data respectively. This also avoids physically sending the data which might have lead to consumption of lots of manpower.

The paltry arrangement of downloading every one of the information and decoding locally is plainly unreasonable, because of the gigantic measure of transmission capacity taken a toll in cloud scale frameworks. Additionally, besides wiping out the neighbourhood stockpiling administration, putting away information into the cloud fills no need unless they can be effortlessly searched and used.

Therefore, investigating protection saving and viable search benefit over encoded cloud information is of central significance. Considering the possibly extensive number of on-request information clients and immense measure of outsourced information records in the cloud, this issue is especially testing as it is to a great degree hard to meet likewise the prerequisites of execution, framework ease of use and versatility.

II. Literature Review

The various works related to keyword based search and the secure dynamic search in the cloud are mentioned and explained in the following part of this chapter. The early works on search in the cloud were mainly focused on single keyword based method, similarity based search [6], [7], multiple keyword based search [2], multiple keyword ranked search [1], [4], [5] and the encryption based search [1], [3]. Some of them are explained briefly in the following section of this chapter.

In 2014, Ning Cao, Cong Wang, Kui Ren and Wenjing Lou the members of IEEE [4] for the first time, they explored the problem of multiple keyword ranked based search in the cloud environment for data retrieval and they established the collection of strict security requirements for such kind of secure cloud information utilization management system.

According to Neelam S. Khan, Dr. C. Rama Krishna and Anu Khurana the members of IEEE in 2014, their work [6] concentrated among the initial couple of ones to investigate multiple-keyword ranked search along with fuzzy search over outsourced encoded information. They plan technique called RFMS (Ranked Fuzzy Multi-Keyword Search) to address the multiple keyword based search issues. In RFMS, it is accepted that the measure of information keeps on expanding now and again. In accordance with that keyword dictionary need to be extended intermittently. They proposed the new enhanced dictionary pattern, presented another index calculation algorithm and utilized a better keyword based searching method that included multiple keyword based searching along with fuzzy technique and created a better coordinating outcome sets.

In 2016, Zhihua Xia, Xinhui Wang, Xingming Sun and Qian Wang the members of IEEE proposed a secure tree-based search method [1] over the scrambled cloud information, which can support multiple-keyword ranked search and dynamic operation on the gathering of documents. Here "Term Frequency Inverse Document Frequency (IDF)" and vector space models are joined for construction of Index and generation of query to support Multiple Keyword Ranking based search.

In [2], they have proposed a multiple-keyword conjunctive query method for different information proprietors in the distributed computing. They explored multiple keyword conjunctive secure query method [2] over scrambled cloud information for various information proprietors situation. They initially develop a protected keyword query skeleton for multi-proprietor model and find that it is to a great degree wasteful to accomplish multiple keyword conjunctive method. To enhance query effectiveness, they build up the query skeleton and proposed a genuine multiple keyword conjunctive query method with practical query proficiency. They hypothetically dissect the execution and security of their proposed method and facilitate tentatively assess the rightness and viability in a genuine informational collection.

According to the survey their evaluation results shows that the index construction in their proposed method is comparatively costly calculation process because of their time-consuming exponentiation calculation in the gathering of composite request. Also it conveys that the time cost of search can be additionally diminished by outsourcing the search operations to the computationally intense cloud eventually. Additionally there is no ranking service using which the documents could have been retrieved at the faster rate. All these issues are considered while proposing the methodology.

III. Methodology

In the proposed scheme Parallel Search Architecture is used as shown in Fig.1. In that Multi-Threading schema for parallel execution of searching operation has been used. Here when user search with the help of multiple keywords, the bulk numbers of search requests are divided into multiple batches. And each batch will be handled separately by different threads.

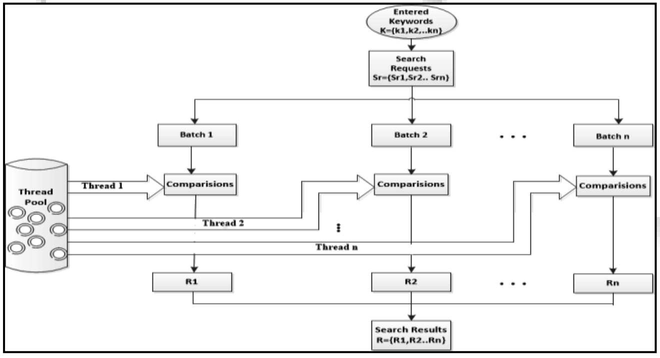


Fig. 1: Architecture of Proposed System

According to this every thread performs following actions simultaneously:

1) It searches by comparing the length of the characters in the keyword matching with the length of characters present in the required keyword present in the cloud.
   For example:
   a) Let the keywords stored in the cloud are like [Nano, Bulk Import, Important, Kernel], at first when we start searching with the key "Import", it will count the number of letters as 6 and starts comparing them with the list of Keywords in cloud, if there is any length difference or the value is zero then it takes to next level with filtered results, otherwise 0(null) results will be produced.
2) Also it simultaneously searches for the perfect matching of all letters (Word matching) of the keyword.
   For example:
   a) When the keywords stored in the cloud are like [Nano, Bulk Import, Important, Kernel], if we start searching with key like "Import", then the threads searches with filtering the keys with producing result like [Bulk Import ,Important] with their respective rankings.

Number of threads to be used is decided by the system on checking the number of search requests arrived. This method improves searching efficiency and reduces time cost for data retrieval.

Next in order to support dynamic searchable encryption, the proposed method converts the keyword at the File owner side to Hashed Keyword using MD5 hash generator scheme and then store it in the cloud. This maintains the keyword information privacy in the cloud storage. Later while retrieving the data or document the Hashed Keywords are internally decrypted and compared with the keywords entered for searching. This enables the dynamic searchable encryption in the cloud.

A. *Overall System Architecture*

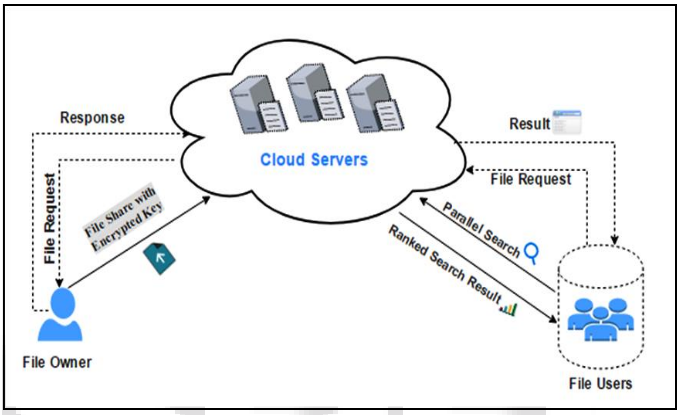


Fig. 2: Overall System Architecture Diagram

Above Fig.2 shows the overall system architecture for data sharing in the cloud environment. In this scenario, at first the file owner uploads the file with the multiple keywords. These keywords are converted to hashed format at the file owner side and linked with the file before uploading to the cloud servers. The next part is to share this file to particular user or group of users. When the file is shared to the particular user or group of users in the cloud storage, the authorized users are allowed to search them in the cloud. Here the users enter the multiple keywords which later converted to hashed format to search their respective documents. In this case, the users make use of parallel search mechanism to search the documents with less time cost. At the cloud side the entered keywords in hashed format are compared with existing keywords and matching files with the proper ranking are provided as result to the users.

There is another case where the cloud users can request the files to the file owners. In this scenario the users enter the multiple keywords and send the request through the cloud. Here also the keywords are converted to hashed format while sending the request. These keywords are compared with the keywords present in the cloud and if match is found then the respective file owners receive the request. Otherwise the request is just rejected in the cloud side only.

In the case of keywords matching, if the file owners are interested to share then he/she can accept the request or else he/she can reject that particular request. Once the file owners accept the request the users are allowed to download the file.

IV. DESIGN & IMPLEMENTATION

A. *Entity-Relationship diagram*

Entity-Relationship model is used to represent all the entities and their respective relationships as an abstract outline. Fig.3 shows the Entity-Relationship diagram for the developed application.

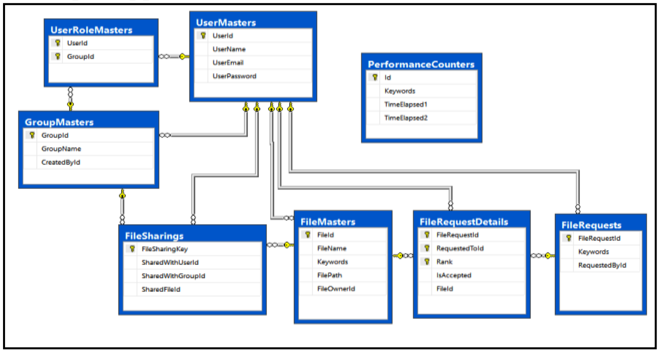


Fig. 3: Entity-Relationship diagram

The entities involved in the database are as follows:

1) UserMasters: This entity involves the registration of the users for using the application by providing their details like UserName, E-mail and password along with the UserId as primary key.
2) GroupMasters: This entity involve the creation of the groups by providing the information like GroupName and the CreatedById field is automatically set to the user's Id whoever is creating the group. Here the GroupId is used as the primary key.
3) UserRoleMasters: This entity involves the assignment of the selected users to the group. Here the UserId and the GroupId is stored as the composite foreign keys.
4) FileMasters: This entity stores the information of the files uploaded by the file owners. It involves the information like Filename, keywords in hashed format, filepath and FileOwnerId. Here the FileId is used as the primary key and FileOwnerId i.e., ID of the file uploaded user is used as the foreign key.
5) FileSharings: This entity stores the information related to files shared by the file owners to other users and groups. This has the fields like FileSharingKey as the primary key, shared files id i.e., SharedFileId, shared with users Id i.e., SharedWithUserId and shared with groups Id i.e., SharedWithGroupId respectively. This has the three foreign keys from the tables like FileMaster, Usermaster and GroupMaster.
6) FileRequests: This entity stores the information regarding the file requested by the users to file owners with the help of keywords. This involves the fields like FileRequestId as the primary key, keywords and RequestedById i.e., the id of the user who requested that file. This also in turn stores the information in the FileRequestDetails tables in the case of entered keywords matching with that of original file's keyword.
7) FileRequestDetails: This entity stores the information of the files requested along with respective File owner information. The values get added to this entity only in the case of entered keywords matching with that of the original file's keyword. This involves the fields like FileRequestId which is used as both primary and foreign key, file owner's id i.e., RequestedToId, FileId as the foreign key along with Rank to check how much percentage do the entered keywords matches with the original one, IsAccepted status to check whether the file is shared by file owner or not. The IsAccepted status is in Boolean format i.e., in the case of file request approved by the file owners this status turns to TRUE or else in other cases it will remain FALSE.
8) PerformanceCounters: This entity stores the information regarding the comparison and calculation of the performance values in the case of with and without using multithreading paradigm while searching the files and requesting the files from the file owners. This has the fields like Id as the primary key, Keywords, TimeElapsed1 i.e., time elapsed for searching and requesting the files without using multithreading concept and TimeElapsed2 i.e., time elapsed for searching and requesting the files with using multithreading concept. Here both the TimeElapsed1 and TimeElapsed2 are obtained in the milliseconds value.

B. *Requirements Implementation*

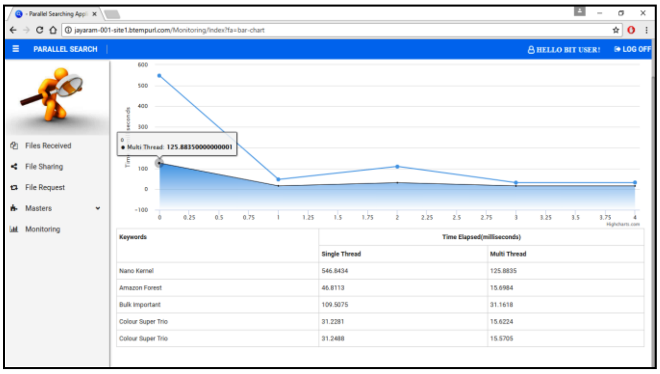
This application is implemented using

1) Programming Language used:
   Back end: ASP .NET MVC (Model View Controller).
   Front end: HTML, Bootstrap, CSS, JavaScript and Json.
2) Software used: Visual Studio 2015.
3) Database Management: Microsoft SQL server management studio 2014
4) Operating System used: Windows 7 OS.

V. EXPERIMENTAL RESULTS

In this section the outcomes and screenshots have been displayed keeping in mind the end goal to exhibit the adaptability and secure data sharing between the multiple users in the cloud environment and searching of this data with the help of multiple keywords ranking based Parallel Search method. Also it demonstrates the different screenshots of the users and super admin and exhibits the usefulness of the application.

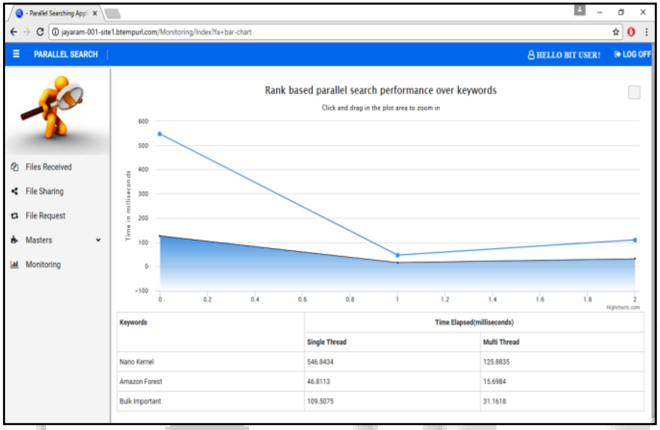
A. *Frontend Results*



Keywords	Time Elapsed(milliseconds)	
	Single Thread	Multi Thread
Nano Kernel	546.8434	125.8835
Amazon Forest	46.8113	15.6984
Bulk Important	109.5075	31.1618
Colour Super Trio	31.2281	15.6224
Colour Super Trio	31.2488	15.5705

Fig. 4: Performance Comparison Graph for Files Search

Fig.4 shows the screenshot of the performance comparison graph when a file is searched by the user. This graph is built on the basis of time taken for searching the files shared by the owner with the help of multiple keywords. Here the X axis indicates the keywords and the Y axis indicates the time taken for the respective keywords during search process for which the units are in milliseconds. In the above figure 6-12 for every keywords two time values i.e., with and without using multithreading is calculated for comparison purpose during search process. Also the respective single thread and multiple thread time values are tabulated below the graph in the same page. So from the values tabulated in the table and graph points we get to know that the time taken for searching the files with using Multithreading process is far lesser than the time taken for searching the files without using the multithreading process.

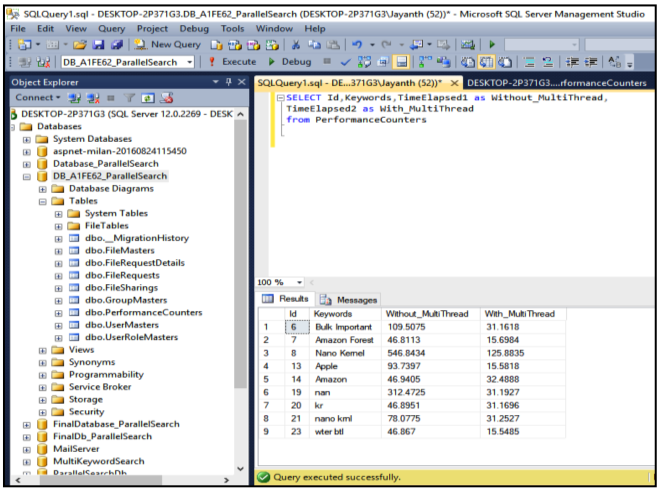


Keywords	Time Elapsed(milliseconds)	
	Single Thread	Multi Thread
Nano Kernel	546.8434	125.8835
Amazon Forest	46.8113	15.6984
Bulk Important	109.5075	31.1618

Fig. 5: Performance Comparison Graph for File Request process

Fig.5 shows the screenshot of the performance comparison graph when a file is requested by the user. This graph is built on the basis of time taken for sending the file request to the owner with the help of keywords. Here the X axis indicates the keywords and the Y axis indicates the time taken for the respective keywords during file requesting process for which the units are in milliseconds. In the above figure Fig.5 for every keywords two time values i.e., with and without using multithreading is calculated for comparison purpose during file requesting process. Also the respective single thread and multiple thread time values are tabulated below the graph in the same page. So from the values tabulated in the table and graph points we get to know that the time taken for sending the file request with using Multithreading process is far lesser than the time taken for sending the file request without using the multithreading process.

B. Backend Results



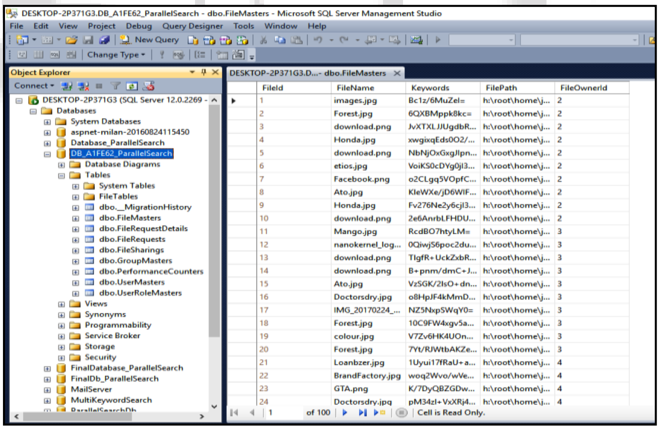
```
SELECT Id,Keywords,TimeElapsed1 as Without_MultiThread,
TimeElapsed2 as With_MultiThread
from PerformanceCounters
```

	Id	Keywords	Without_MultiThread	With_MultiThread
1	6	Bulk Important	109.5075	31.1618
2	7	Amazon Forest	46.8113	15.6984
3	8	Nano Kemel	546.8434	125.8835
4	13	Apple	93.7397	15.5818
5	14	Amazon	46.9405	32.4888
6	19	nan	312.4725	31.1927
7	20	kr	46.8951	31.1696
8	21	nano kml	78.0775	31.2527
9	23	wter btl	46.867	15.5485

Query executed successfully.

Fig. 6: Performance Values

Above Fig.6 shows the screenshot of the Performance values obtained in the database at the backend during the searching and file requesting process. Here the performance values are measured in the milliseconds of time. As shown in the Fig.6 we can compare the values for time taken to search the documents with the help of multiple keywords by using with and without multithreading schema. Here we can see that the time taken to search the documents by using multithreading schema is far lesser than the time taken to search the documents without using multithreading schema. Hence it shows that the proposed method i.e., Parallel Search improves searching efficiency and reduces time cost for data retrieval.



FileId	FileName	Keywords	FilePath	FileOwnerId
1	images.jpg	Bc1z/6MuZel=	h:\root\home\j...	2
2	Forest.jpg	6QXBMppk8kc=	h:\root\home\j...	2
3	download.png	JvXTXLJJUgdbR...	h:\root\home\j...	2
4	Honda.jpg	xwgixqEds0O2/...	h:\root\home\j...	2
5	download.png	NbNjOxGxgJlpn...	h:\root\home\j...	2
6	etios.jpg	VoiKS0cDYg0jl3...	h:\root\home\j...	2
7	Facebook.png	o2CLgq5VOpfC...	h:\root\home\j...	2
8	Ato.jpg	KleWXe/jD6WIF...	h:\root\home\j...	2
9	Honda.jpg	Fv276Ne2y6cjl3...	h:\root\home\j...	2
10	download.png	2e6AnrbLFHDU...	h:\root\home\j...	2
11	Mango.jpg	RcdBO7htyLM=	h:\root\home\j...	3
12	nanokernel_log...	0QiwjS6poc2du...	h:\root\home\j...	3
13	download.png	TlgfR+UckZxbR...	h:\root\home\j...	3
14	download.png	B+pnm/dmC+J...	h:\root\home\j...	3
15	Ato.jpg	VzSGK/2lsO+dn...	h:\root\home\j...	3
16	Doctorsdry.jpg	o8HpJF4kMmD...	h:\root\home\j...	3
17	IMG_20170224_...	NZ5NxpSWqY0=	h:\root\home\j...	3
18	Forest.jpg	10C9FW4xgv5a...	h:\root\home\j...	3
19	colour.jpg	V7Zv6HK4UOn...	h:\root\home\j...	3
20	Forest.jpg	7Yt/RJWtbAKZe...	h:\root\home\j...	3
21	Loanbzer.jpg	1Uyui17fRaU+a...	h:\root\home\j...	4
22	BrandFactory.jpg	woq2Wvo/wVe...	h:\root\home\j...	4
23	GTA.png	K/7DyQBZGDw...	h:\root\home\j...	4
24	Doctorsdry.jpg	pM34zl+VxXRj4...	h:\root\home\j...	4

1 of 100 | Cell is Read Only.

Fig. 7: File Details in Backend

Above Fig.7 shows the screenshot of the File Master information obtained in the database at the backend during the file uploading process. Here it can be seen that the keyword information is stored in the encrypted format. This proves that the keywords attached to file while uploading to the cloud database will be in secure form and it will be impossible for the unauthorized person to retrieve the data from the cloud. Finally it is shown that this allows the dynamic searchable encryption in the cloud.

VI. CONCLUSIONS & FUTURE WORK

At present the cloud innovation has gotten comfortable and most utilized innovation everywhere throughout the globe. So recovering the documents via searching securely in the cloud has offered ascend to numerous security and time cost concerns. There are a few techniques to search the documents and retrieving it securely and with less time cost. The proposed parallel search scheme is one of such scheme.

As shown in the backend result screenshot of Chapter V i.e., Fig 6, for the records of 100 the time taken to search the required file with the help of keywords like “Bulk Important” is just 31 milliseconds whereas without the usage of the proposed method it would have taken double or thrice the time obtained using the proposed method.

So it offers following advantages:

1) Reduces the time cost to search the documents.
2) Provide the efficient and secure multiple-keyword and ranking based search over the cloud.

In present proposed application there finds a limitation in the file size to be uploaded and transferred. In future this issue should be redressed keeping in mind the end goal to further build the compelling utilization of this application.

REFERENCES

[1] Zhihua Xia, Member, IEEE, Xinhui Wang, Xingming Sun, Senior Member, IEEE, and Qian Wang, Member, IEEE, “A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data.” Vol.27, No.2, February 2016.

[2] Hui Yin∗†, Zheng Qin, Jixin Zhang, Wenjie Li, Lu Ou, Yupeng Hu Keqin L,“Secure Conjunctive Multi-keyword Search for Multiple Data Owners in Cloud Computing”, 2016 IEEE 22nd International Conference on Parallel and Distributed Systems.

[3] Russell W. F. Lai, Sherman S. M. Chow, Department of Information Engineering, “Structured Encryption with Non-Interactive Updates and Parallel Traversal”, 2015, IEEE 35th International Conference on Distributed Computing Systems.

[4] Ning Cao, Member, IEEE,Cong Wang, Member, IEEE, Ming Li, Member IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE, “Privacy-Perserving Multi-Keyword Ranked Search over Encrypted Cloud Data”, Vol.25, No.1, January 2014.

[5] Neelam S. Khan, Dr. C. Rama Krishna, Anu Khurana, “Secure Ranked Fuzzy Multi-Keyword Search over Outsourced Encrypted Cloud Data”, 2014, 5th International Conference on Computer and Communication Technology.

[6] Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu, Department of Computer Science, “Efficient Similarity Search over Encrypted Data”, 2015, IEEE 28th International Conference on Data Engineering.

[7] Cong Wangy, Kui Reny, Shucheng Yux, and Karthik Mahendra Raje Ursy, Department of ECE and CS, “Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data”, 2012 IEEE INFOCOM.

[8] Flora Amato, Francesco Gargiulo, Antonino Mazzeo, Sara Romano, Carlo Sansone Dipartimento di Ingegneria Elettrica e delle Tecnologie dellInformazione, “A semantic search engine in the cloud”, 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.