

CONTENTS

1. ABSTRACT

2. INTRODUCTION

2.1 Objective

2.2 Proposed System

2.3 Review of Literature

3. PROBLEM DEFINATION AND SCOPE

3.1 Definitions, Acronyms, and Abbreviations

4. HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements

4.2 Software Requirements

5. PROJECT PLAN

6. DESIGN AND IMPLEMENTATION

6.1 Design

6.2 System Architecture

6.3 Implementation

7. SCREENSHOTS

8. CONCLUSION

9. REFERENCES

1. ABSTRACT

Using cloud computing, individuals can store their data on remote servers and allow data access to public users through the cloud servers. As the outsourced data are likely to contain sensitive privacy information, they are typically encrypted before uploaded to the cloud. This, however, significantly limits the usability of outsourced data due to the difficulty of searching over the encrypted data.

In this project, we address this issue by developing the multi-keyword search schemes over encrypted cloud data. First, we introduce the relevance scores and preference factors upon keywords which enable the precise keyword search and personalized user experience. Second, we develop a practical and very efficient multi-keyword search scheme.

This technique makes use of index building, trapdoor generating and query processing. Lastly, we analyse the security of the proposed schemes in terms of confidentiality of documents, privacy protection of index and trapdoor, and unlinkability of trapdoor. The proposed scheme can achieve high security level comparing to the existing ones and better performance in terms of functionality, query complexity and efficiency.

2. INTRODUCTION

The cloud computing treats computing as a utility and leases out the computing and storage capacities to the public individuals. In such a framework, the individual can remotely store her data on the cloud server, namely data outsourcing, and then make the cloud data open for public access through the cloud server. This represents a more scalable, low-cost and stable way for public data access because of the scalability and high efficiency of cloud servers, and therefore is favorable to small enterprises. The outsourced data may contain sensitive privacy information. It is often necessary to encrypt the private data before transmitting the data to the cloud servers. The data encryption, however, would significantly lower the usability of data due to the difficulty of searching over the encrypted data.

The Project provides a very easy solution to enable efficient search over encrypted files. The basic concept behind this project is to generate several keywords over outsourced data. These keywords are later encrypted and stored over the cloud server. When the search user needs to access the outsourced data, it can select some relevant words. These keywords are mapped onto the index file and returns files containing matching results to the search user.

2.1 Objective

1. Improve usability of outsourced data.
2. To develop technique to remove difficulty in searching over encrypted data.
3. To retrieve more precise result using relevance score and preference factor of keywords in search keyword set.

2.2 Proposed System

Our proposed system, increases usability of outsourced data by enabling efficient search over encrypted files and provides accurate relevance score calculation between encrypted index and query vectors.

2.3 Review of Literature

2.3.1 Secure and privacy preserving keyword search:

Qin Liu in this paper proposed that the search that provides keyword privacy, data privacy and semantic secure by public key encryption. CSP is involved in partial decipherment by reducing the communication and computational aerial in decryption process for end users. The user submits Keyword trapdoor encrypted by user's private key to CS (Cloud Server) securely and retrieves the encrypted documents.

Limitations:

Greater communication cost for encryption and decryption.

2.3.1 Single Keyword Search Over Encrypted data on cloud:

Obtainable searchable encryption scheme consent to a user to firmly look for over encrypted data through keywords without first applying decryption on it, the proposed techniques support only conventional Boolean keyword search, without capturing any applicability of the files in the search result. When directly applied in large joint data outsourcing cloud environment, they go through next shortcoming. These schemes can only return the results which match all the keywords simultaneously and cannot rank the returned results.

Limitations:

- Single-keyword search without ranking.
- Boolean- keyword search without ranking.
- Do not get relevant data.

3. Problem Definition and Scope

The multi keyword search over encrypted data enables to design and develop a multi keyword ranked search technique over outsourced data to enhance search efficiency.

The project is designed to overcome the drawbacks of existing system like single-keyword search without ranking, Boolean- keyword search without ranking single-keyword search with ranking.

To overcome these drawbacks we have proposed searchable encryption to enable searching over encrypted cloud data. We introduce the relevance scores and the preference factors of keywords for searchable encryption. The relevance score can enable more precise returned results and the preference factors represent the importance of keywords in the search keyword set specified by search users and correspondingly enables personalized search to cater to specific user preferences. Finally it returns the document with the most matching keywords.

3.1. Definitions, Acronyms, and Abbreviations

1. User: Any user (Search user, Data Owner, Cloud Server)
2. RAM: Random Access Memory
3. Search User: Any user that user from the cloud server.
4. Cloud Server: Intermediate entity which stores the encrypted documents and corresponding indexes
5. Data Owner: Outsources data to the cloud for convenient and reliable data access

4. HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements:

- Setup of public cloud provided by Amazon.
- Minimum 500 MB Hard disk space.
- Minimum 515 MB RAM

4.2 Software Requirements:

- Setup of public cloud provided by Amazon.
- Java virtual machine (JVM)
- Net Beans IDE 8.0.2 for designing the front end
- My SQL Server 5.0 for backend.
- Eclipse-jee-juno-SR1

5. PROJECT PLAN

Sr. No	Action	Start Date	End Date
1	Literature Survey	1/8/17	07/8/17
2	Synopsis	15/8/17	22/8/17
3	SRS	23/8/17	29/8/17
4	Collecting data sets	30/8/17	5/9/17
5	Creating user registration and login page	6/9/17	10/7/17
6	Pre-processing datasets	11/9/17	22/9/17

6. DESIGN AND IMPLEMENTATION

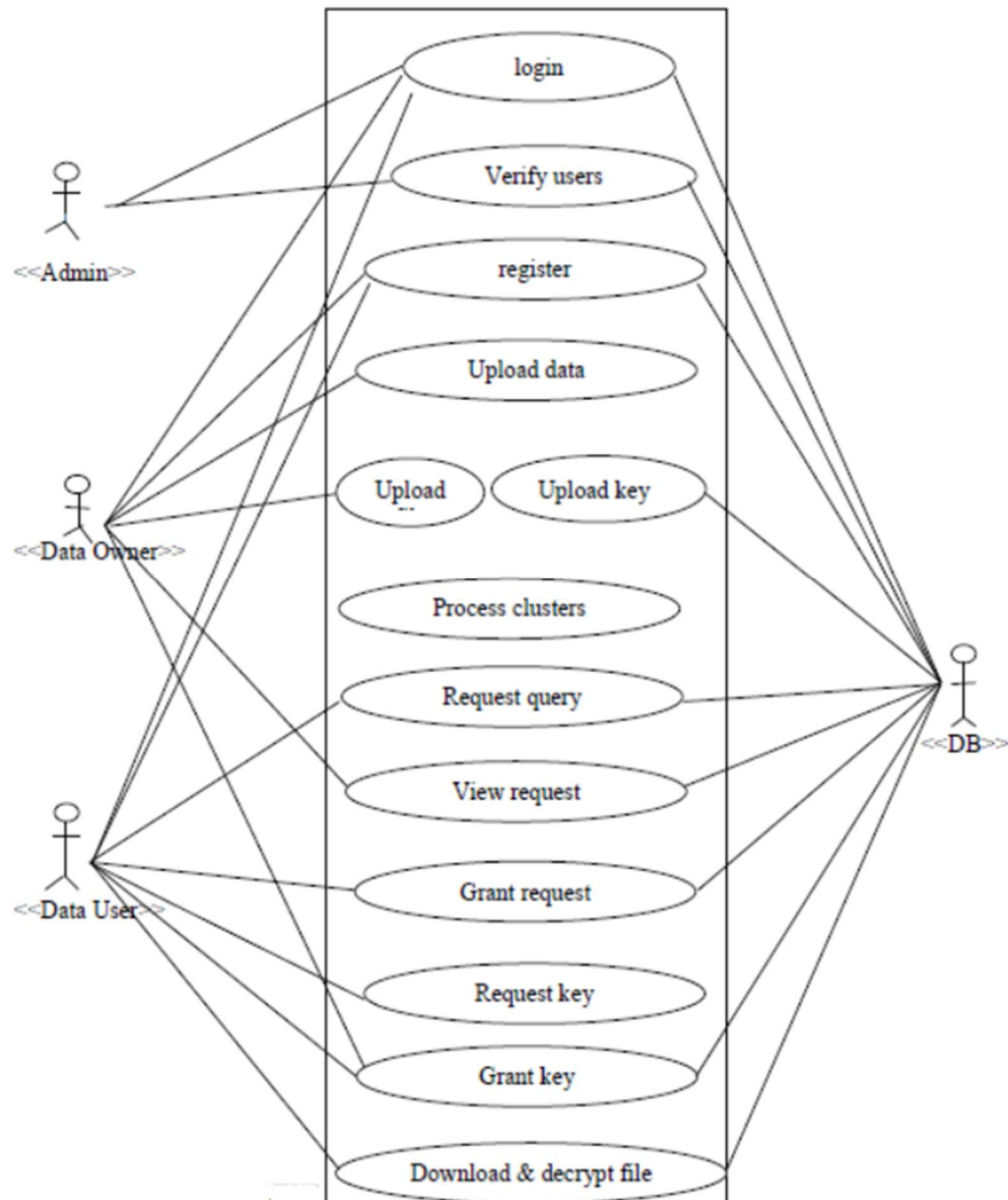
6.1 Design

Design is the meaningful engineering representation of something that is to be built. It can be traced to a customer's requirements and at the same time assessed for quality against a set of predefined criteria for "good" design. Design Strategy is a high level statement about the approach to develop a system. In other words it can be described as a particular approach to develop a system. It includes statements on the system's functionality, hardware and system software platform, and method for acquisition. Design requires experimentation. It is generally divided into two steps:

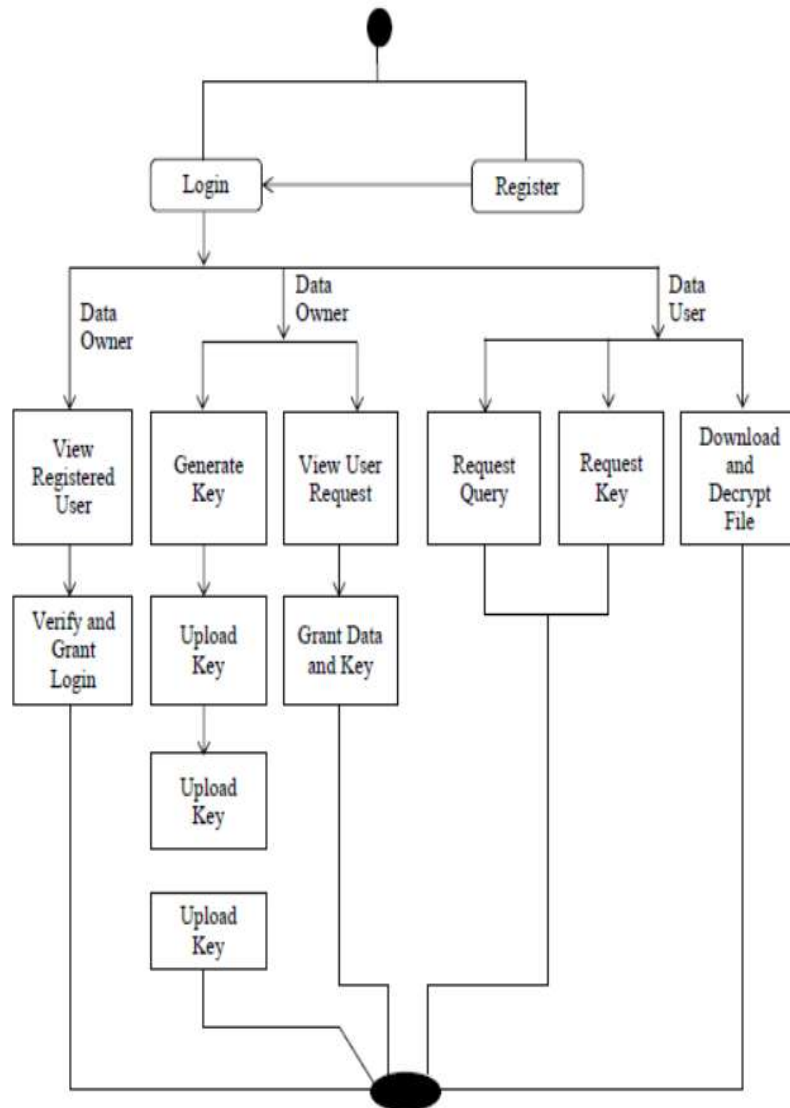
1. A logical step that is independent from the implementation environment.
2. A physical step that focuses on the ordering of resources and details pertaining to programming languages or to the execution environment.

The development of an application may be divided into several major areas. They are chained sequentially within a waterfall lifecycle, or they are distributed among the various iterations of an iterative lifecycle. In the project used Object Oriented Design Strategy as the approach for designing the software. Object Oriented Design (OOD) creates a representation of the real-world problem domain and maps it into a solution domain that is the software. OOD results in a Design that interconnects data objects and processing operations in a way that modularizes information and processing rather than processing alone.

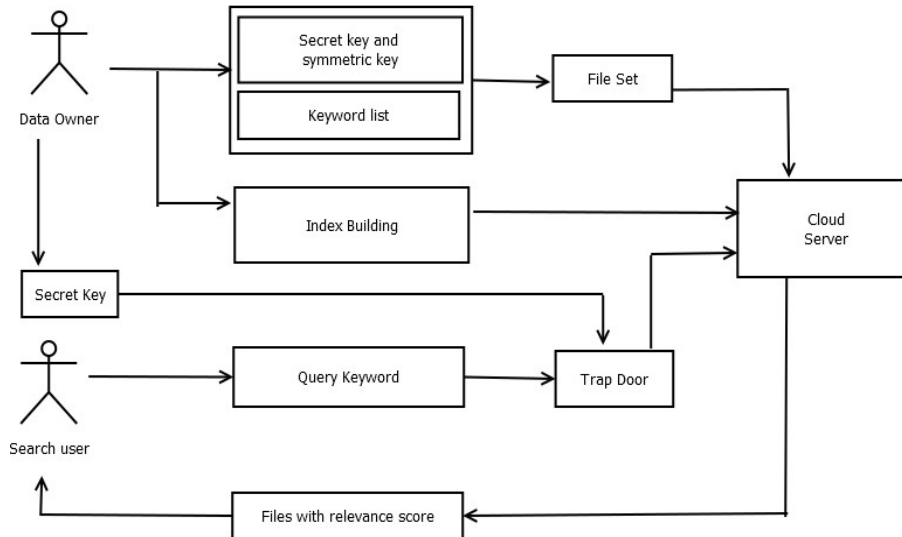
6.1.1 Use case diagram



6.1.2 Activity Diagram



6.2 System Architecture



6.3 Implementation

In our proposed system, the first procedure is initialization in which keywords are generated automatically from the input file set and index is formed by mapping the keywords to the respective files. The second procedure is encryption of file set and index and upload of these two onto the cloud server.

The next procedure involves query processing, where the trapdoor is generated from the query fired by the search user. With the help of trapdoor and index, the search user gets the matching file set in descending order. Finally the file chosen by the search user is decrypted using key provided by the data owner.

Following algorithms are used:

1. Key Generation

Secret key $K=(S, M_1, M_2)$ is generated where S is a $(m+1)$ -dimensional binary vector M_1 and M_2 are two $(m+1)*(m+1)$ invertible matrices

The data owner sends private key k_1 and public key k_2 to search users through a secure channel.

2. Index Building

The data owner first utilizes encryption algorithm to encrypt the document collection $F = (F_1, F_2, \dots, F_N)$ with the private key K_1 .

The encrypted document collection are denoted as C_j ($j = 1, 2, \dots, N$)

P is an m -dimensional binary vector according to C_j , where each bit $P[i]$ indicates whether the encrypted document contains the keyword w_i , i.e.

$P[i] = 1$ indicates 'YES' i.e. the encrypted document contains keyword and

$P[i] = 0$ indicates 'NO' i.e. the keyword is not present in the document.

P is extended to a $(m + 1)$ -dimensional vector P' , where $P'[m + 1] = 1$.

Vector S (randomly generated binary vector) functioning as a splitting indicator, splits P' into two $(m + 1)$ -dimensional vectors (P_a, P_b) .

Namely,

If $S[i] = 0$ ($i = 1, 2, \dots, m + 1$), $P_a[i]$ and $P_b[i]$ are both set as $P'[i]$;

if $S[i] = 1$ ($i = 1, 2, \dots, m + 1$), the value of $P'[i]$ will be randomly split into $p_a[i]$ and $P_b[i]$ where $(P'[i] = P_a[i] + P_b[i])$.

The index of encrypted document C_j is calculated as $I_j = (P_a * M_1, P_b * M_2)$.

Where M_1 and M_2 are invertible matrices and $S = M_1 * M_2$.

Finally, the data owner sends $C_j || FID_j || I_j$ ($j = 1, 2, \dots, N$) to the cloud server. Index of encrypted document C_j is calculated as $I_j = (P_a * M_1, P_b * M_2)$.

2. Trapdoor Generation

The search user generates the keyword set W for searching.

Q is an m -dimensional binary vector according to W where $Q[i]$ indicates whether the i^{th} keyword of dictionary w_i is in W , i.e., $Q[i] = 1$ indicates yes and $Q[i] = 0$ indicates no. Q is extended to a $(m + 1)$ -dimensional vector Q' , where $Q'[m + 1] = -s$.

Now, the search user chooses a random number $r > 0$ to generate $Q'' = r * Q'$ and Q'' is split into two $(m + 1)$ vectors (Q_a, Q_b) .

If $S[i] = 0 (i = 1, 2, \dots, m + 1)$, the value of $Q'' [i]$ will be randomly split into $Q_a[i]$ and $Q_b [i]$;

if $S[i] = 1 (i = 1, 2, \dots, m + 1)$, $Q_a[i]$ and $Q_b[i]$ are both set as $Q''[i]$.

Thus, the search trapdoor $T (W)$ can be generated as $(M_1^{-1} Q_a, M_2^{-1} Q_b)$.

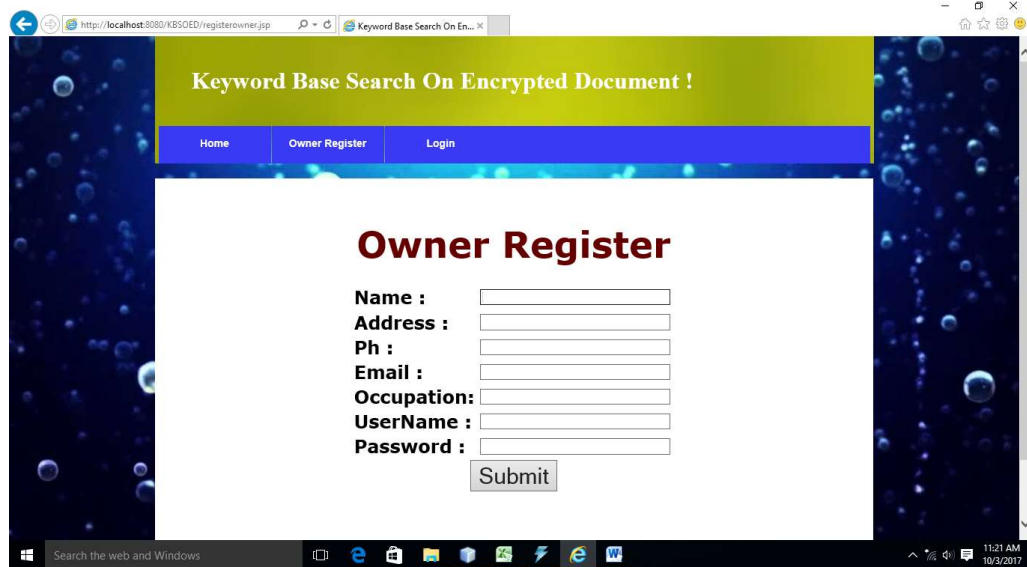
4. Query

With the index $I_j (j = 1, 2, \dots, N)$ and trapdoor $T(W)$, the cloud server calculates the query result as

$$R_j = I_j * T_W = (P_a * M_1, P_b * M_2) * (M_1^{-1} * Q_a, M_2^{-1} * Q_b) = P_a * Q_a + P_b * Q_b = P' * Q'' = r * P' * Q' = r * (P * Q^{-s}).$$

If $R_j > 0$, the corresponding document identity FID_j will be returned.

7. Screenshots



The screenshot shows a web browser window with the URL `http://localhost:8080/KBSOED/registerowner.jsp`. The page has a yellow header with the text "Keyword Base Search On Encrypted Document !". Below the header is a blue navigation bar with links for "Home", "Owner Register", and "Login". The main content area is white and features the title "Owner Register" in bold red text. Below the title are input fields for "Name", "Address", "Ph", "Email", "Occupation", "UserName", and "Password". A "Submit" button is located at the bottom of the form. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock indicates 11:21 AM on 10/3/2017.

Keyword Base Search On Encrypted Document !

Home Owner Register Login

Owner Register

Name :

Address :

Ph :

Email :

Occupation:

UserName :

Password :

Submit

Fig 7.1: Registration Frame



The screenshot shows a web browser window with the URL `http://localhost:8080/KBSOED/login.jsp`. The page has a yellow header with the text "Keyword Base Search On Encrypted Document !". Below the header is a blue navigation bar with links for "Home", "Owner Register", and "Login". The main content area is white and features the title "Log In" in bold red text. Below the title are input fields for "UserName" and "Password". A "Submit" button is located at the bottom of the form. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock indicates 11:21 AM on 10/3/2017.

Keyword Base Search On Encrypted Document !

Home Owner Register Login

Log In

UserName :

Password :

Submit

Fig 7.2: Log-In Frame

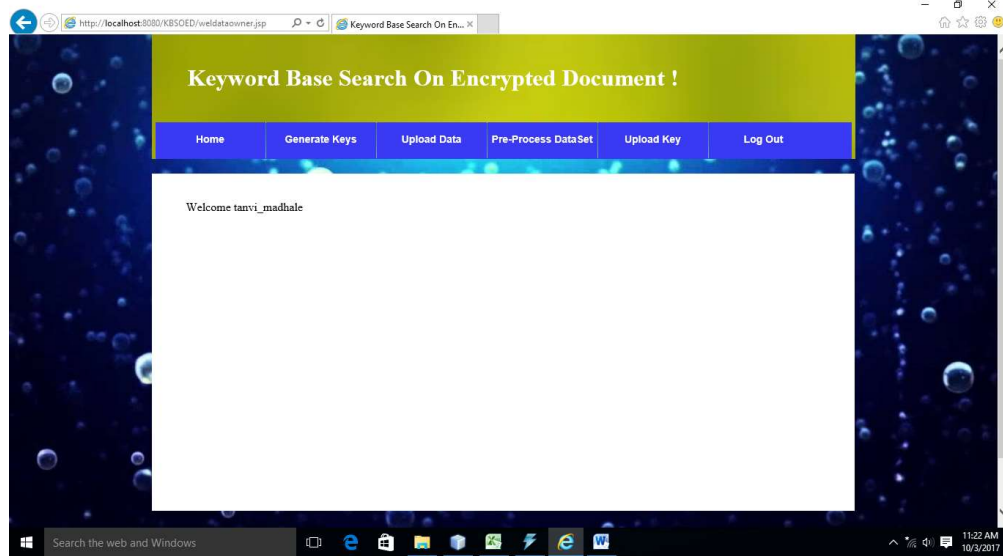


Fig 7.3: Data Owner Personal Frame

8. CONCLUSION

We presented a model that enables searching over encrypted data in a more efficient manner. The index building approach is used to reduce the search overhead and computational cost in terms of time. The trapdoor function guarantees calculation of accurate relevance score and preference factor. The Query processing module returns the set of files with the matching keywords.

The relevance score is associated with every keyword in the file set. The Term Frequency (TF) and Inverted Document Frequency (IDF) are calculated. The Term Frequency (TF) describes the probability of that keyword to be occurred in the respective file. The Inverted Document Frequency (IDF) describes the probability of that keyword to be occurred in the set of files.

Our proposed system enhances and overcomes limitations of traditional single-keyword search and thus reduces computational cost in terms of time.

9. REFERENCES

1. Cong Wang, Ning Cao, KuiRenand Wenjing Lou. "Enabling Secure and Efficient Keyword Search over Outsourced Cloud Data". IEEE Transaction on Parallel and Distributed Systems, VOL. 23, NO. 8; August 2012.
2. D. Song, D. Wagner, and A. Perrig. "Practical techniques for searches on encrypted data" in Proc. of IEEE Symposium on Security and Privacy; 2000, pp. 44-55.
3. "Secure Indexes for Searching Efficiently on Encrypted Compressed Data"
E. J. Goh. Technical Report 2003/216, Cryptology Print Archive, <http://eprint.iacr.org/2003>.
4. Boneh D, Crescenzo G, Ostrovsky R, Persiano G. "Public Key Encryption with Keyword Search", In: Proceedings of Eurocrypt 2004, Lecturenotes in computer.
5. Singhal A. "Modern information retrieval: A brief overview", IEEE Data Engineering.
6. Kevin Hamlen, Murat Kantarcioglu, Security Issues for cloud computing, in International Journal of Information Security and Privacy, 2010.

