PRIVACY PRESERVING RANKED MULTI-KEYWORD SEARCH FOR MULTIPLE DATA OWNERS IN CLOUD COMPUTING

A PROJECT REPORT

Submitted by

RAJAN KUMAR [1031310587]
ANURAG ROY [1031310583]

Under the guidance of
Ms. R.YAMINI
(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2017

SRM UNIVERSITY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**PRIVACY PRESERVING RANKED MULTI-KEYWORD SEARCH FOR MULTIPLE DATA OWNERS IN CLOUD COMPUTING**" is the bonafide work of " **RAJAN KUMAR [1031310587], ANURAG ROY [1031310583]** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. R.YAMINI
GUIDE
Assistant Professor
Dept. of CSE

SIGNATURE

Dr. B. AMUTHA
HEAD OF THE DEPARTMENT
Dept. of CSE

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

In this modern age, cloud computing is more popular. Cloud computing gives many opportunity to the user. Through this opportunity user can upload their data and download that data from the cloud. It also gives the function so that user can run many websites by pay-per-use service. Most cloud server do not serve single user, it gives service to multiple user at the same time. This project consist of functions in which user can search multiple file and share the file to multiple user. It has ranking search technique in which most frequent searches are shown. This project also contain the function of key generation. Through this a dynamic secret key is generated which prevent others from stealing data.

ACKNOWLEDGEMENTS

We would like to place on record, our grateful thanks to **Dr.T.R. PACHAMUTHU**, chancellor and **PROFESSOR PRABIR K. BAGCHI**, vice chancellor, for providing all facilities and help in carrying out this project. We also would like to thank **Dr.C. MUTHAMIZHCELVAN**, Director, Engineering and Technology for the stimulus provided.

We would also like to thank **Dr. B. Amutha, Professor and Head of Department**, Computer Science and Engineering, SRM University for providing us conductive ambience for developing our project. We would also like to thank our B.Tech Project Coordinators, **Dr.Revathi Venkataraman Professor, Mrs.B.Ida Seraphim and Mrs.G.Abirami, Assistant Professor**, Department of Computer Science and Engineering, SRM University for continuous support.

We would also like to express our gratitude to my project guide **Ms. R. Yamini, Assistant Professor**, and Class Incharge **Ms.A.Nithya Kalyani**, Assistant Professor, Department of Computer Science and Engineering, SRM University for his assistance, timely suggestions and guidance throughout the duration of this project. We also convey our gratitude to panel head **Dr. M Murali** , Assistant Professor and to all other members in the project panel and those who have contributed to this project directly or indirectly.

Author

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AJAX -> Asynchronized Java Script and XML

HTML -> Hyper Text Markup Language

J2EE -> Java 2 Enterprise Edition

JAVA -> A Programming Language

JRE -> Java Runtime Environment

JSP -> Java Server Page

JVM -> Java Virtual Machine

SQL -> Structured Query Language

XML -> Extensible Markup Language

CHAPTER 1

INTRODUCTION

Cloud computing becomes popular and has important role in our lives. Cloud computing gives many benefits such as data storage, running websites on the cloud. Through these benefits user can upload their data and download that data from the cloud. Users can upload many documents like financial details, government details etc. All these information can be access by the user. To secure the privacy in cloud, the files has to be encrypted first before uploading to cloud and after that the data is decrypted by the key while downloading.

In this project, when the user wants to search a particular file then the cloud server can perform search without knowing the exact keyword. This searching technique gives rank-wise result to the user. This ranking technique is based on the frequently search the file name by the user.

It also contain the encryption and decryption technology. Encryption is done at the time of uploading the data to cloud. Advance Encryption Standards methodology is used to file encryption. Decryption is done at the time of downloading the data from cloud. Data Encryption Standards methodology is used to file decryption. User have the key to decrypt data by using decryption key.

It also have sharing function. When the user wants to share the file to others then a key is generated. This key is generated to protect the system from attackers. By using this key, user can download data from cloud.

CHAPTER 2

LITERATURE SURVEY

2.1 Cryptographic Solution to a Problem of Access Control in a Hierarchy

2.1.1 ABSTRACT

A plan in light of cryptography is proposed for get to control in a framework where chain of command is spoken to by an in part requested set (or poset).

Clear usage of the plan requires clients very set in the chain of importance to store a substantial number of cryptographic keys. A period versus-capacity exchange off is then portrayed for tending to this key administration issue.

2.2 Dynamic and Efficient Key Management for Access Hierarchies

2.2.1 ABSTRACT

The issue of key administration in a get to pecking order has evoked much enthusiasm for the writing. The pecking order is displayed as an arrangement of in part requested classes can acquire classes of her class through key deduction. Answer for the above issues have properties:

(i) Use its own key to determine a descendants key.

(ii) Secure data at a class comprises of a solitary key related with that class.

(iii) Refreshes (revocations, increments, and so on.) are dealt with locally in the chain of command.

(iv) This is probably secure from attacker.

2.3 EXISTING SYSTEM

In the Existing System, the system have encryption and decryption method. Through these methods encryption and decryption process are done. In this system, the data owner share file to a single user and the user who is authorized can download the shared files by giving the decryption key.

The Existing System does not have any searching technique of the data files. This system was not secured because it does not uses any secret channel to send the key for file decryption. So it is very much prone to be attacked by the hackers.

It also takes more time to share files to user because it only serves a single user at one time. It is less scalable and less efficient.

2.3.1 DISADVANTAGES

1. It does not have searching process.
2. It does not have multiple sharing technique.
3. It takes more time to share the files.
4. The cost is high.
5. It is less scalable.
6. It is less efficient.

2.4 PROPOSED SYSTEM

In this paper, we proposes scheme which enables users for searching files by giving the keyword. This project consist of protocols in which users uses different keys to encrypt and decrypt files. When the file is uploaded in the cloud then the data file is encrypted. Advance Encryption Standards methodology is used to file encryption. While downloading data from cloud then the data files are decrypted. Data Encryption Standards methodology is used to file decryption. User have the key to decrypt data by using decryption key.

It consist of secure search protocol in which user can search files by giving the short or full data files names. The search gives the frequently result. It allow the searches over the encrypted data files. Secret key is send via a secure channel to prevent it from attacking. This key is generated at the time, when the file is shared. This protocol is used to send the decryption key to the authenticated shared file users. This proposed system is secure. It uses Dropbox to store the data files on the cloud and uses Gmail to send the key to single/multiple users.

2.4.1 ADVANTAGES

1. It allow searches over encrypted files.
2. Decryption key should be send via a secure channel like Gmail.
3. Data files are encrypted on the cloud server.
4. The cost is low.
5. It is more efficient.
6. It is more scalable.
7. It is more secured.
8. It is more reliable.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Hardware Requirements

Processor	:	Intel i3
HDD	:	50 GB
Display	:	15 VGA
RAM	:	1 GB

3.2 Software Requirements

OS	:	Windows
Coding Language	:	Java
Design Language	:	XML
Front End	:	Net beans 7.0
Back End	:	SQLYog
Database	:	MYSql

3.3 Functional Requirements

An functional requirement record characterizes the usefulness of a framework or one of its subsystems. It likewise relies on the kind of programming, expected clients and the sort of framework where the product is utilized. Functional user necessities might be abnormal state explanations of what the framework ought to do yet useful framework prerequisites ought to likewise portray plainly about the framework benefits in detail.

CHAPTER 4

SYSTEM DESIGN

System configuration is the way toward characterizing the engineering, segments, modules, interfaces and information for a framework to fulfill determined necessities. System configuration could be viewed as the utilization of frameworks hypothesis to item advancement.

4.1 Activity Diagram

An activity diagram outwardly shows a progression of activities or stream of control in a framework like a flowchart or an information stream graph. Movement charts are frequently utilized as a part of business process displaying. They can likewise portray the means in an utilization case graph. Exercises demonstrated can be successive and simultaneous. In both cases a movement graph will have a start and an end.

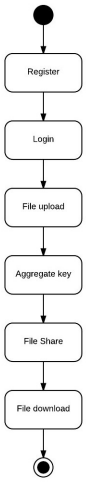


Figure 4.1: Activity Diagram

4.2 Use-Case Diagram

Use case graphs are normally alluded to as conduct outlines used to depict an arrangement of activities (utilize cases) that some framework or frameworks (subject) ought to or can perform in a joint effort with at least one outside clients of the framework (on-screen characters). Each utilization case ought to give some perceptible and significant outcome to the performing artists or different partners of the framework.

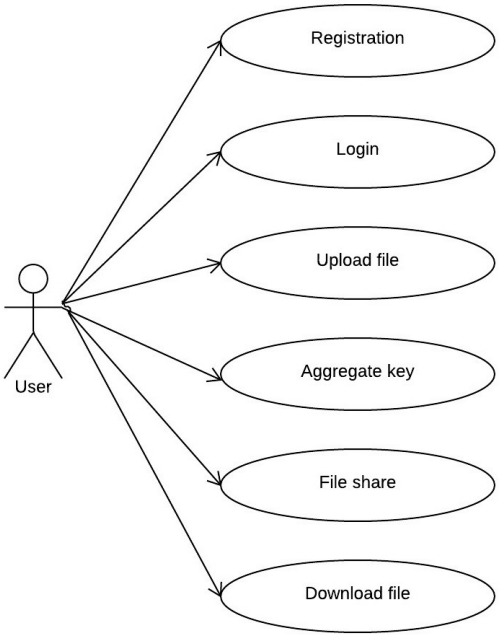


Figure 4.2: Use Case Diagram

4.3 Data Flow Diagram

A data flow diagram (DFD) shows how information is handled by a framework regarding data sources and yields. As its name demonstrates its attention is on the stream of data, where information originates from, where it goes and how it gets put away.

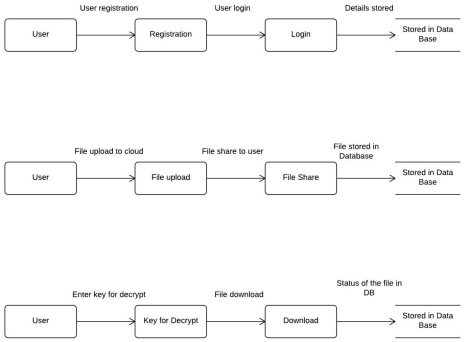


Figure 4.3: Data Flow Diagram

4.4 Class Diagram

It speaks to the static perspective of an application. Class diagram is not just utilized for imagining, depicting and recording diverse parts of a framework additionally to construct executable code of the product application.

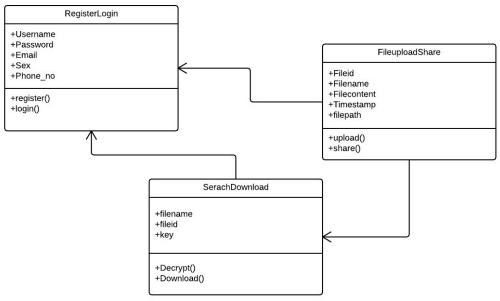


Figure 4.4: Class Diagram

4.5 E-R Diagram

An entity relationship diagram (ERD) is an information demonstrating procedure that graphically delineates a data framework elements and the connections between those elements.

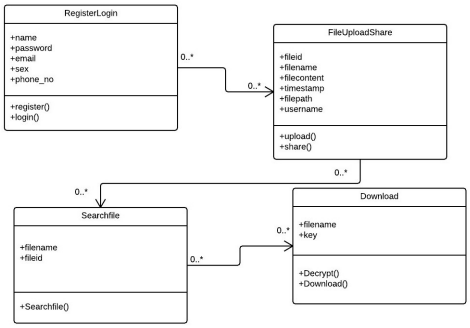


Figure 4.5: E-R Diagram

4.6 State Diagram

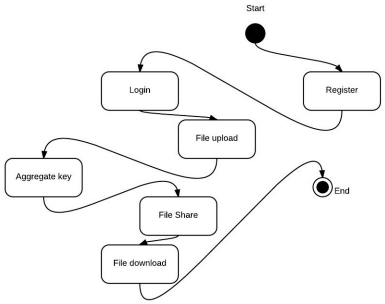


Figure 4.6: State Diagram

CHAPTER 5

SOFTWARE ENVIRONMENT

5.1 Introduction

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The primary language is JAVA. In this project HTML is also being used for UI.

5.2 The Java Programming Language

It is a high level language that is described by following keywords:

1. Easy to use
2. Object oriented
3. Portable
4. High performance

5.3 The Java Platform

A platform is the equipment or programming condition in which a program runs. We have specified probably the most famous stages same as Windows, Linux and MacOS. The stage varies components used. The Java stage has two segments:

1. Java Virtual Machine (Java VM)
2. Java Application Programming Interface (Java API)

Virtual Machine support like pillars in the stage of Java and portable into different type of equipment stages. Application Programming Interface is a huge accumulation

instant of programming segments which gives numerous valuable abilities, for example, graphical UI (GUI) gadgets. Application Programming Interface used into libraries to support the program for better results. This figure shows that Application Programming

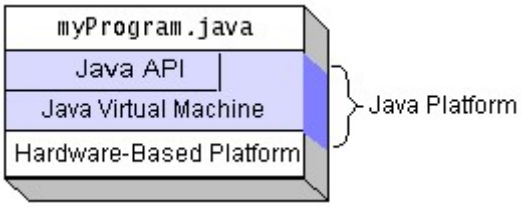


Figure 5.1: Java platform

Interface and Virtual Machine protect program from components.

5.4 JDBC

JDBC is developed by Sun Microsystem for Java Database Connectivity. JDBC offers a generic SQL database access medium. JDBC was first made public in March of 1996, it was released for 90 days public developer review. After insider's review the final JDBC was released soon after.

CHAPTER 6

PROBLEM FORMULATION

In this section, we propose description of system architecture and the security goals of our project which is to be achieved.

6.1 PROBLEM DEFINITION

This project is used to search multiple keywords and share files to multiple users. It consist of searchable technique which gives the facility to search over cloud.

6.2 ARCHITECTURE MODEL

The proposed system contain data users, the cloud server and the administration server. When the users have right to access in the server from their local system, they select the file which have to upload on the cloud.

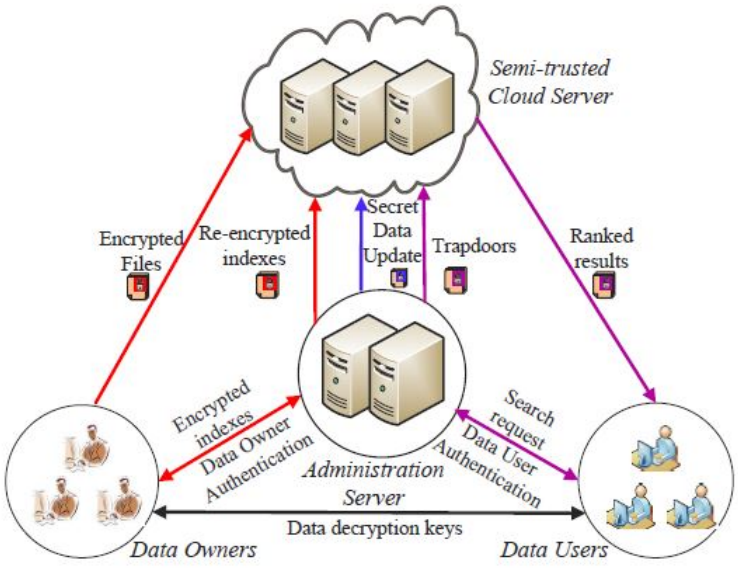


Figure 6.1: Architecture model

This selected files is saved on the location where all the users files are saved (Because there is not only single user, there are multiple user which cannot interact directly). Now from that location, the user uploads the data file on the cloud. When the data is uploaded, if the user wants to share that uploaded files, he/she can share with the registered users along with the private key. When the data files are uploaded on the cloud, the server cannot know the contents of data files because the data files are encrypted.

When the shared users wants to access the file, he/she requests the decryption key. Then the decryption key is sent to their registered mail account. By using that decryption key, the contents of the data file gets decrypted and the users can download the data file.

If any attackers can access the cloud server then he/she cannot get the contents of the actual data files. The cloud server is only responsible for storage of data files.

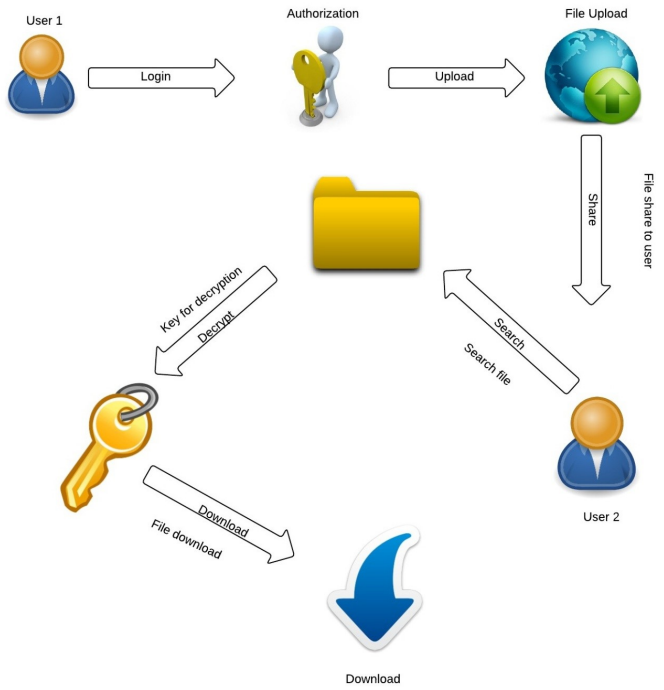


Figure 6.2: Architecture Diagram

6.3 SECURITY GOALS

In this paper, we proposes scheme in which function design satisfy security goals.

6.3.1 Multi Keyword Search over Multiple Data Owner

This paper allow multiple search over encrypted data files. This allows the server to ranked the searched result among different users and return the most frequent results. User can search multiple file names.

This proposed system also search the fuzzy keyword which means if the data file name is rajan.txt and the user search raj then it shows all the data files that contain the letter raj simultaneously. For example, it will display raj.txt, raja.txt, rajesh.txt etc.

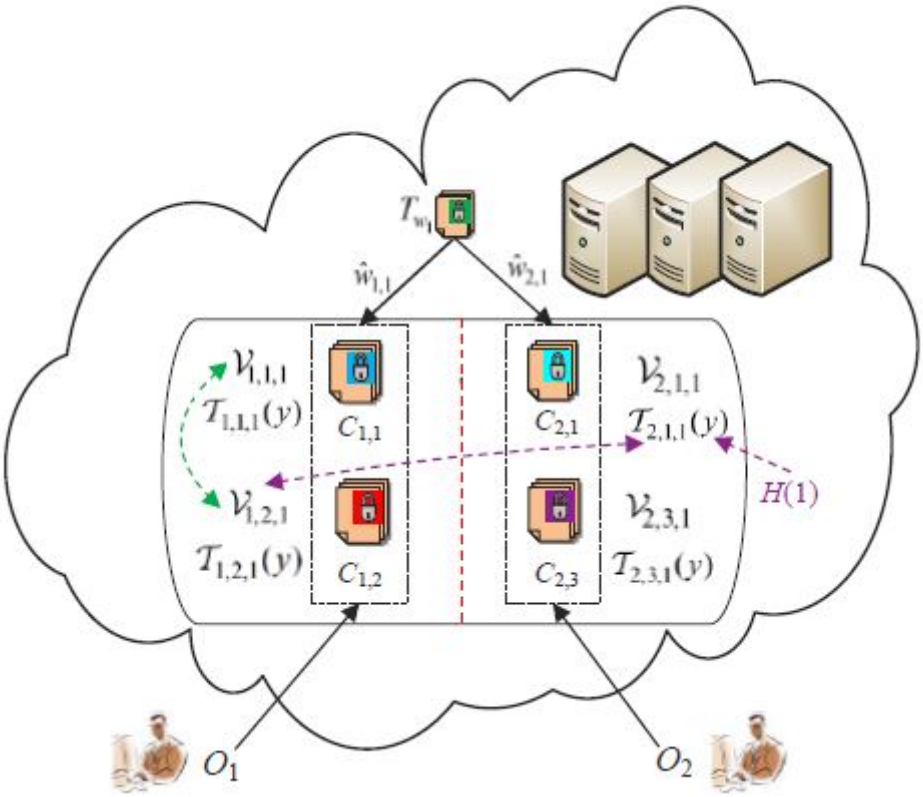


Figure 6.3: Example of ranked search result

6.3.2 User Scalability

This system allow new data users to register and login to the system without disturbing other existing users. At the time of registration, the user gives all the information about himself to the system. All these informations are stored in the database. After the registration is completed, user can login into the system.

6.3.3 User Revocation

This system allow that only registered data users can perform right search over cloud.

6.3.4 Security Process

In this project, we can protect the cloud from being attacked by attacker through eavesdropping. When the files is shared which is in the encrypted form, another users wants the keywords to decrypt the data files.

This key is sent to the user through a secret channel like Gmail. So the attacker cannot access the decryption key. Here SMTP (Simple Mail Transfer Protocol) is used to send the secret key to the shared file users.

6.3.5 Multiple Keyword Search

In this system, user can search multiple file names. This proposed system also search the fuzzy keyword which means if the data file name is abcd.txt and the user search ab then it shows all the data files that contain the letter ab simultaneously. For example, it will display abc.txt, abs.txt etc.

6.3.6 Secret Key Generation

In this system, secret key is generated. For each time of sharing data files, new key is generated and send to the shared users for decrypting the data files to their respective authenticated accounts.

CHAPTER 7

IMPLEMENTATION

7.1 SYSTEM MODULES

1. Authorization and Authentication
2. Uploading and Downloading
3. File Sharing
4. Key Generation
5. Admin Module

7.1.1 Authentication and Authorization

In this module, each and every user have to be registered. First, the user has to provide information about himself/herself. The user have to provide his User-Id, Password, Email-Id, Telephone number, Gender. All these data gets stored in the database. After the registration gets completed, the user can login into the system with the User-Id and the Password. No unauthorized user is allowed to access into the system. This makes the system more secure.

7.1.2 Uploading and Downloading

In this module, Files are uploaded to the server after file is encrypted by the encryption method. This encryption is done by AES (Advanced Encryption Standard) Algorithm and generate key. This Encrypted Data is in the form of Binary and stored in Cloud. User needs decryption key to download the data files. This decryption key is sent to the user through a secret channel. With this key he/she has to decrypt the file and then he/she can download.

7.1.3 File Sharing

In this module, the uploaded files are shared to the multiple users by the data owners. In this system, the Private Key which is to be shared is sent through a secure channel like Gmail. The protocol used to send this secret key is SMTP(Simple Mail Transfer Protocol). The main advantage of this module is that a single file can be shared with multiple users without disturbing other users. This decryption key is used by the user when he is downloading the files.

7.1.4 Key Generation

In this module, when the user wants to access the data files then the server send the decryption key. The user who wishes to access the file receives this key in his authorized Gmail account. He/she then decrypts the file with the help of this key which is generated at the time of file sharing and the file is ready to be downloaded. This key is transferred to the users through a secret channel so that no unauthorized person can access the key and thus protects the system from attackers.

7.1.5 Admin module

This module is mainly gives us information about the data owners, users and files. In this module we can view the users and we can see the files that are uploaded and shared by them. We can also view the list of files and also their status that is whether they are downloaded by the shared user or not.

CHAPTER 8

CODING

8.1 Registration

```
package Registerandlogin;
import DBConnection.DBConnect;
import java.io.*;
import java.sql.*;
import javax.servlet.*;
public class Register extends HttpServlet
{
	protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
	{
		if(password.equals(con_password))
		{
			Connection con=DBConnect.getCon();
			st=con.createStatement();
			int add=st.executeUpdate("insert into user values (’"+username +"’,’"+email+"’,’"+password+"’,’"+gender+"’,’"+phone+"’)");
			if(add>0)
			{
				request.setAttribute("msg","Sucessfully Registered");
				RequestDispatcher rd= request.getRequestDispatcher("login.jsp");
				rd.forward(request, response);
			}
			else
			{
```

```
                    request.setAttribute("msg","Register failed! try again");
                    RequestDispatcher rd= request.getRequestDispatcher("register.jsp");
                    rd.forward(request,response);
                }
            }
            else
            {
                request.setAttribute("msg","ur password and confirm password does not match !");
                RequestDispatcher rd= request.getRequestDispatcher("register.jsp");
                rd.forward(request,response);
            }
        }
}
```

8.2 Login

```
package Registerandlogin;
import DBConnection.DBConnect;
import java.io.*;
import java.sql.*;
import javax.servlet.*;
public class Login extends HttpServlet
{
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        Connection con=DBConnect.getCon();
        st=con.createStatement();
        rs=st.executeQuery("select * from user where name='"+username+"' and password='"+password+"'");
        if(rs.next())
```

```
            {
                RequestDispatcher rd= request.getRequestDispatcher("home.jsp");
                HttpSession session=request.getSession();
                session.setAttribute("username",username);
                rd.forward(request, response);
            }
            else
            {
            RequestDispatcher rd= request.getRequestDispatcher("login.jsp");
            request.setAttribute("msg","Username or password incorrect !");
            rd.forward(request,response);
            }
        }
}
```

8.3 Cloud Manipulation

```
package store;
import com.dropbox.client2.*;
import java.awt.Desktop;
import java.io.*;
import java.net.*;
import java.util.*;
public class Cloudmanupulation
{
        private static final String APP_KEY = "exm9b21pm455n9t";
        private static final String APP_SECRET = "kq0wq7xdy5e17ez";
        private static final AccessType ACCESS_TYPE = AccessType.APP_FOLDER;
        AppKeyPair appKeys = new AppKeyPair(APP_KEY, APP_SECRET);
        public static void main()
        {
            WebAuthSession session = new WebAuthSession(appKeys, ACCESS_TYPE);
```

```
            WebAuthInfo authInfo = session.getAuthInfo();
            RequestTokenPair pair = authInfo.requestTokenPair;
            String url = authInfo.url;
            System.out.println("URL Value "+authInfo.url);
            try
            {
                try
                {
                    Desktop.getDesktop().browse(new URL(url).toURI());
                }
                catch (URISyntaxException ex)
                {
                    Logger.getLogger(Cloudmanupulation.class.getName())
.log(Level.SEVERE, null, ex);
                }
            }
            catch (IOException ex)
            {
                Logger.getLogger(Cloudmanupulation.class.getName())
.log(Level.SEVERE, null, ex);
            }
            JOptionPane.showMessageDialog(null, "Press ok to continue once you have
authenticated.");
            session.retrieveWebAccessToken(pair);
            AccessTokenPair tokens = session.getAccessTokenPair();
            System.out.println("Access granted");
        }
}
```

8.4 Upload To Cloud

```
package com.upload;
import DBConnection.DBConnect;
import java.io.*;
import javax.servlet.*;
import org.apache.commons.fileupload.*;
public class Upload1 extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
        if(ServletFileUpload.isMultipartContent(request))          {
                try {
                        List<FileItem> multiparts = new ServletFileUpload(new DiskFileIt-
emFactory()).parseRequest(request);
                        for(FileItem item : multiparts){
                        if(!item.isFormField()){
                        name = new File(item.getName()).getName();
                        item.write( new File(UPLOAD_DIRECTORY + File.separator + name));
                        String path_name=UPLOAD_DIRECTORY+name;
                        ServletContext s=getServletContext();
                        String FSepa=(String)File.separator;
                        s.setAttribute("FPath", "UPLOAD_DIRECTORY");
                        s.setAttribute("FName", "name");
                        s.setAttribute("Owner", "Admin");
                        Java_Encrypt1 j=new Java_Encrypt1();
                        path=UPLOAD_DIRECTORY;
                        fname=name;
                        Connection con=DBConnect.getCon();
                        Statement st1=con.createStatement();
                        ResultSet rs1;
                        rs1=st1.executeQuery("select filename from upload
```

```
where filename='"+fname+"'");
                    String result="";
                    while(rs1.next())
                    {
                        result=rs1.getString(1);
                    }
                    if(result.equals(fname))
                    {
                        request.setAttribute("message", " sorry! File name already ex-
ist");
                    }
                    else
                    {
                        full_path=UPLOAD_DIRECTORY+name;
                    String uname=(String)
session.getAttribute("username");
                    session.setAttribute("fname", fname);
                    int i=st.executeUpdate("insert into upload (filename, username, filepath,
filekey,filecontent) values ('"+fname+"','"+username+"','"+full_path+"','"+
key_value+"','"+j1+"')");
                    session.setAttribute("filecontent", "filecontent");
                    request.setAttribute("message", "File Uploaded Successfully");
                    }
                }
            }
            }
            catch (Exception ex)
            {
                request.setAttribute("message", "File Upload Failed due to " + ex);
            }          }else{
                request.setAttribute("message","Sorry this Servlet only handles file upload
request");
```

```
}
```

8.5 Send Mail

```
package com.upload;
import DBConnection.DBConnect;
import java.sql.*;
import javax.mail.*;
public class SendMail {
        String host = "smtp.gmail.com";
        String from="rajan_sankarprasad@srmuniv.edu.in";
        props.setProperty("mail.transport.protocol","smtp");
        props.setProperty("mail.host",host);
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.port", "587");
        props.put("mail.smtp.socketFactory.port", "465");
        props.put("mail.smtp.socketFactory.fallback", "false");
        props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        Session mailSession = Session.getInstance(props, new javax.mail.Authenticator()
        {
                protected PasswordAuthentication getPasswordAuthentication()
                {
                        return new PasswordAuthentication("rajan_sankarprasad@srmuniv.edu.in",
"******");
                }
        });
}
```

8.6 Admin

```
package Adminlogin;
import DBConnection.DBConnect;
import java.io.*;
import java.sql.*;
import javax.servlet.*;
public class Adminlogin extends HttpServlet
{
	protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
	{
		Connection con=DBConnect.getCon();
		st=con.createStatement();
		rs=st.executeQuery("select * from admin where Name='"+name+"' and pass-
word='"+password+"'");
		if(rs.next())
		{
			RequestDispatcher rd= request.getRequestDispatcher("adminhome.jsp");
			HttpSession session=request.getSession();
			session.setAttribute("aname", name);
			rd.forward(request, response);
		}
		else
		{
			RequestDispatcher rd= request.getRequestDispatcher("admin.jsp");
			request.setAttribute("msg", "login failed! Try again!");
			rd.forward(request, response);
		}
	}
}
```

CHAPTER 9

SCREENSHOTS

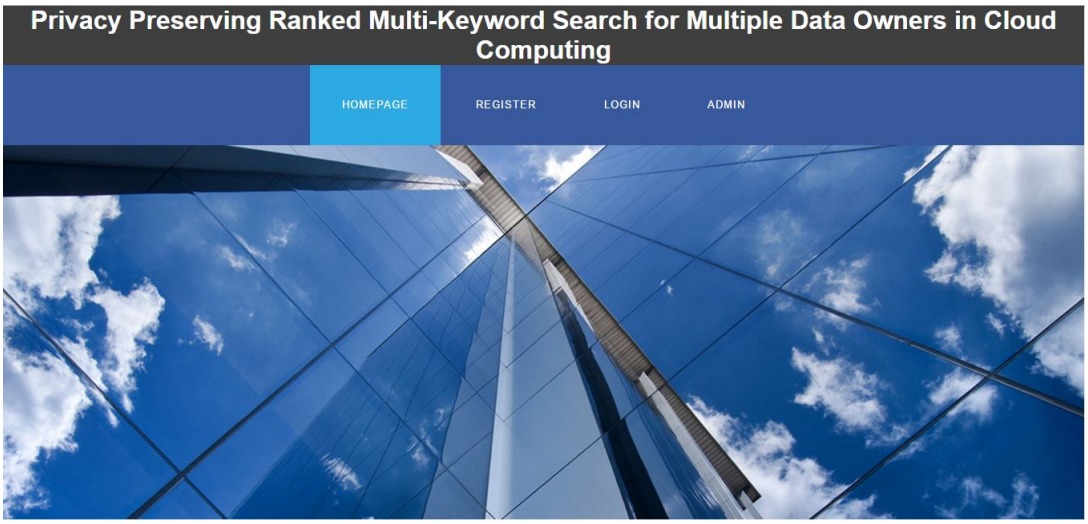


Figure 9.1: HomePage

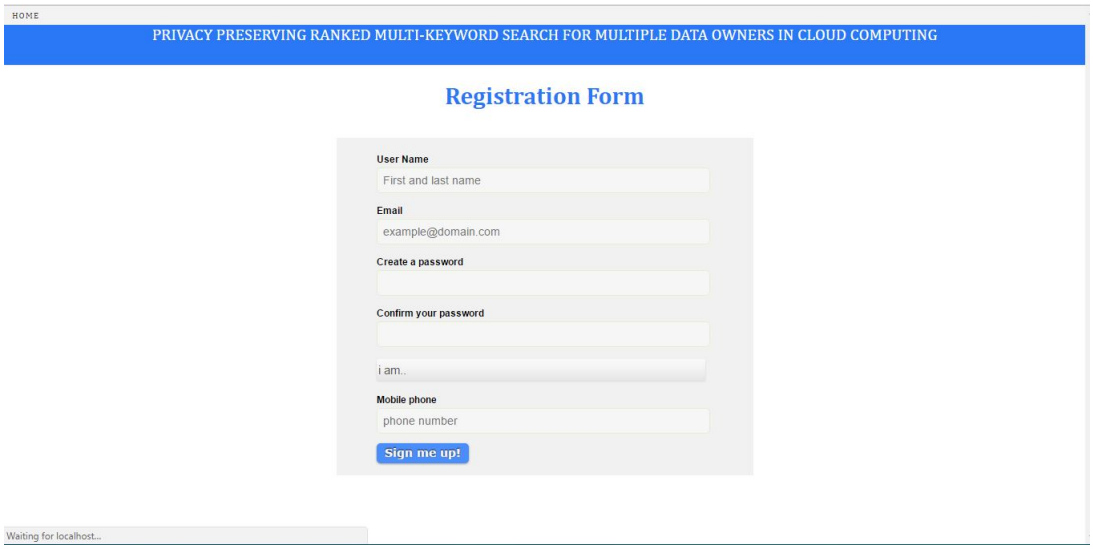


Figure 9.2: Registration Page

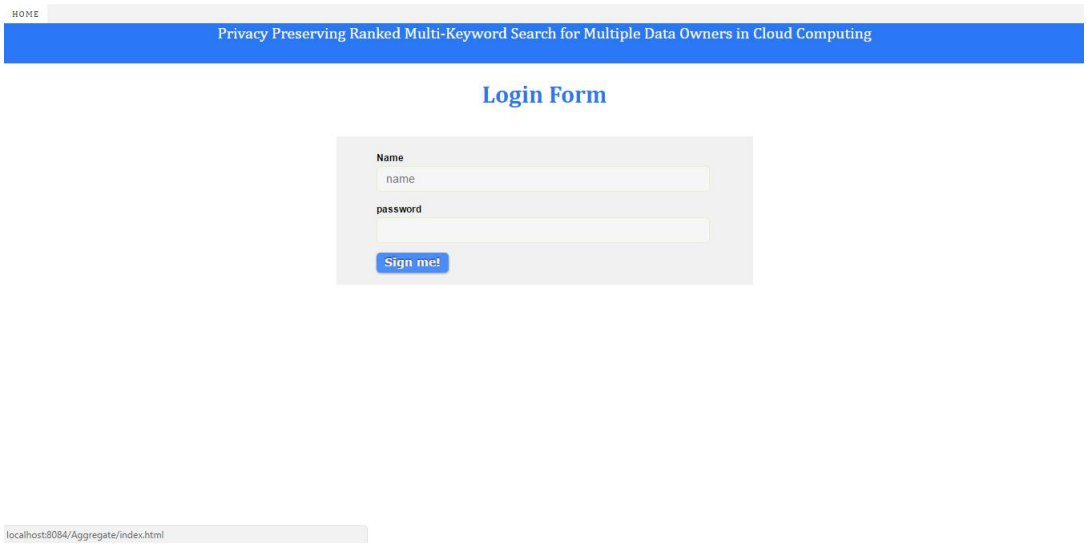


Figure 9.3: Login Page

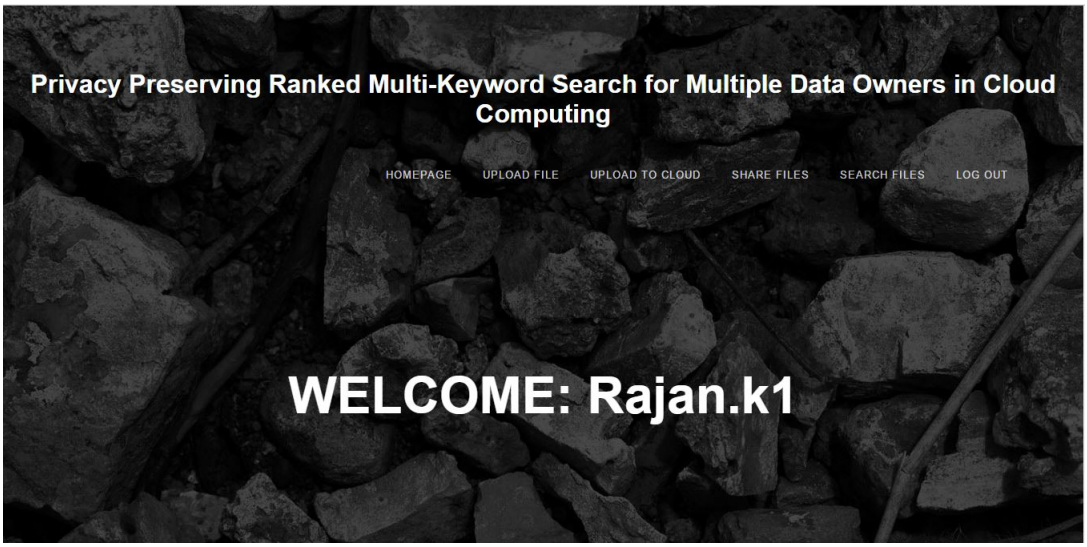


Figure 9.4: UserHome Page

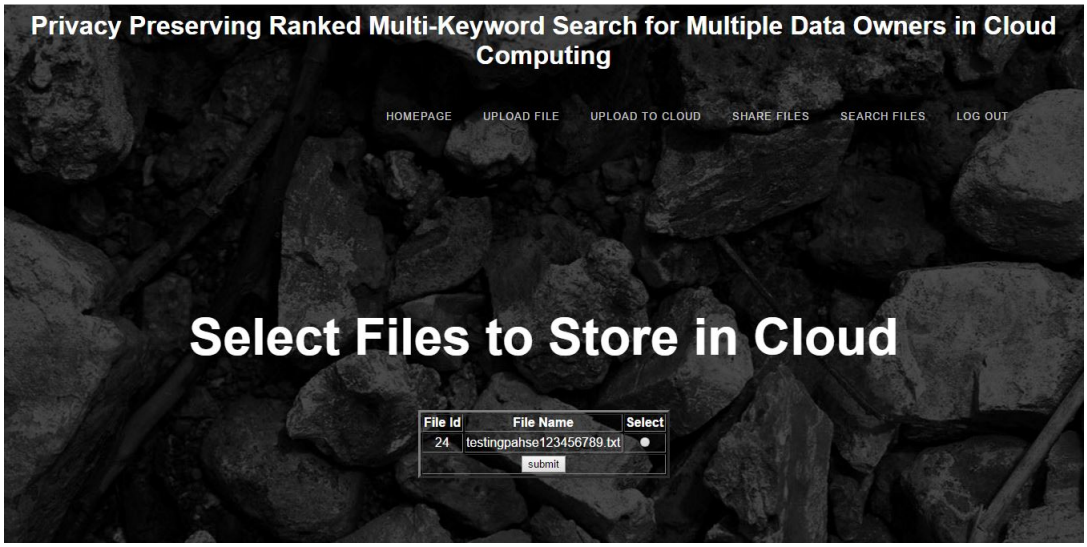


Figure 9.5: upload to Cloud

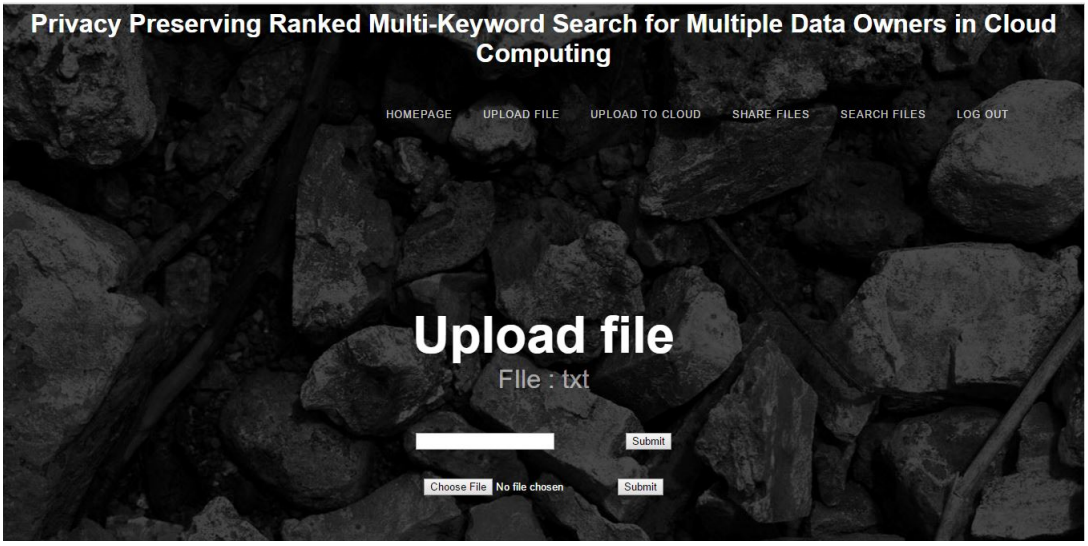


Figure 9.6: Upload File to Cloud

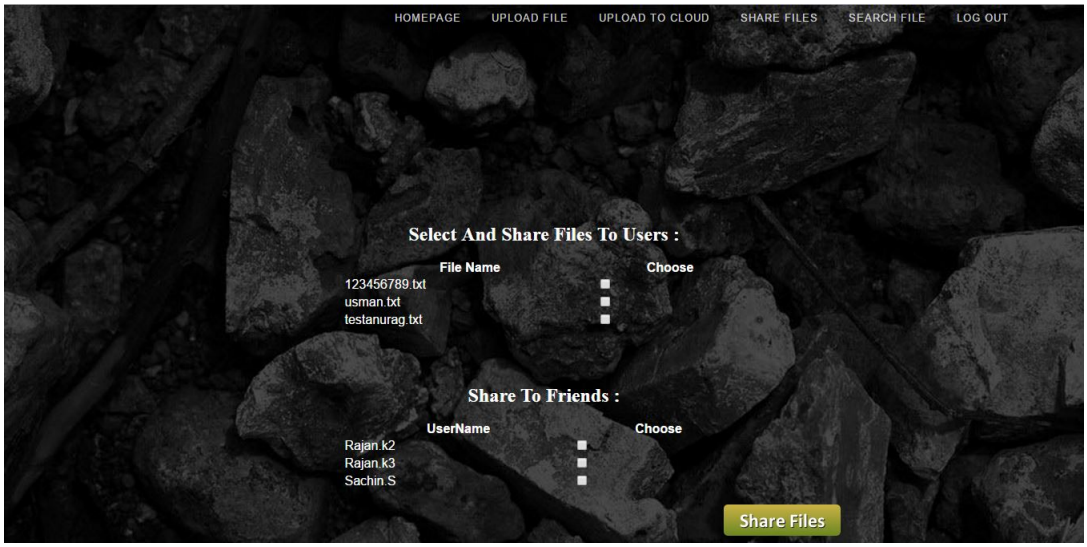


Figure 9.7: Share File between Users

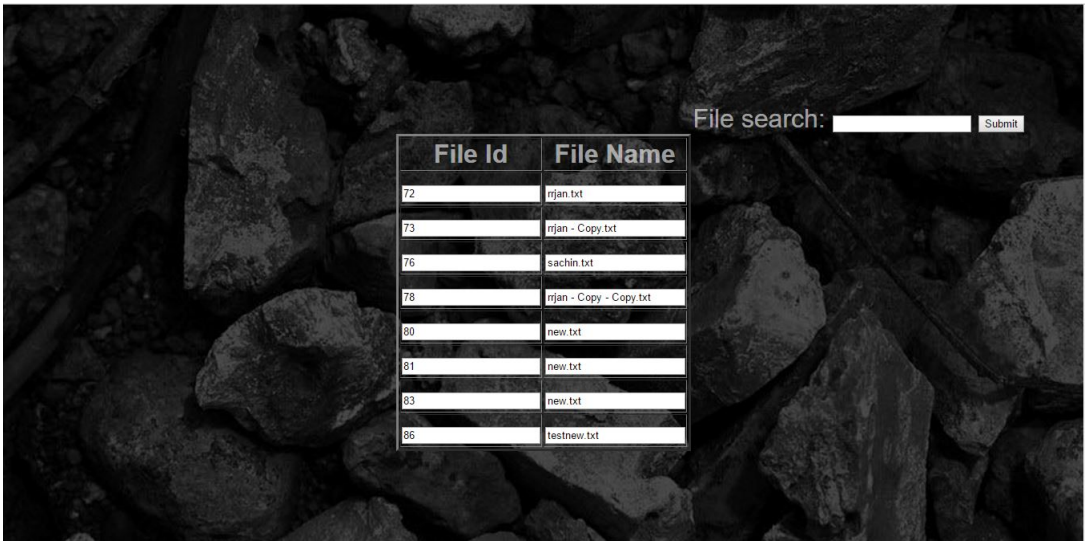


Figure 9.8: Search File

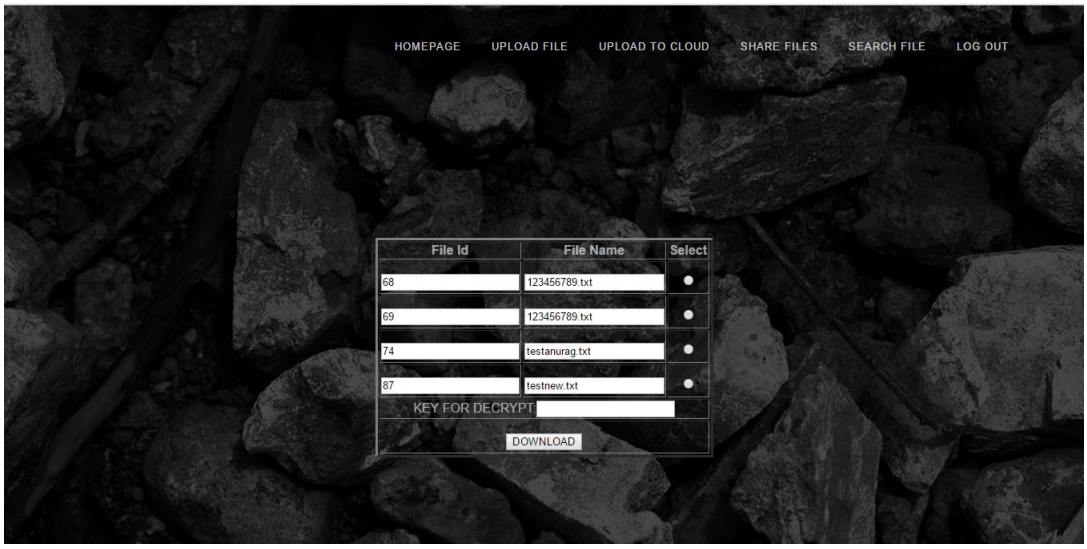


Figure 9.9: Download Shared File

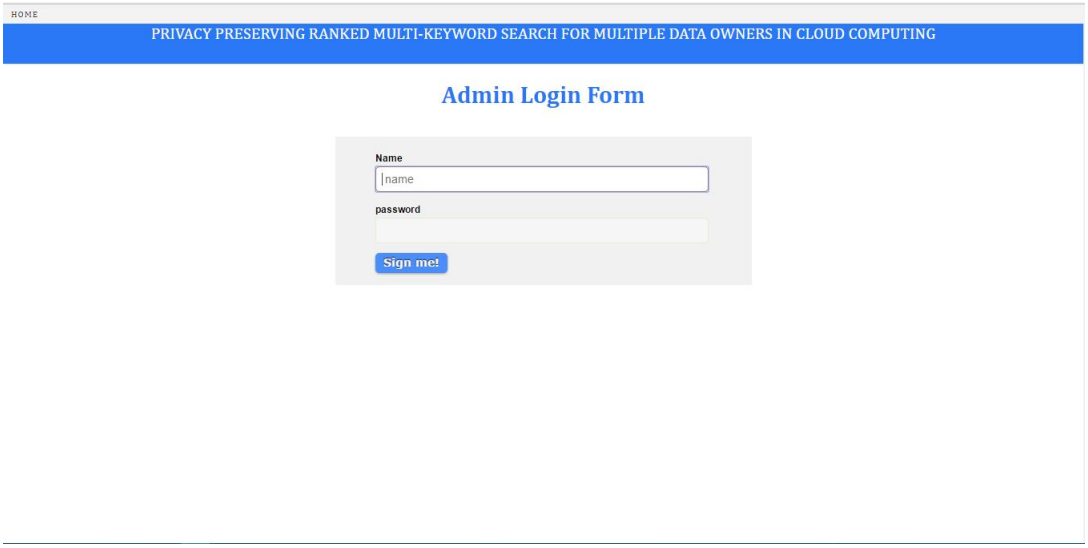


Figure 9.10: Admin Login Page

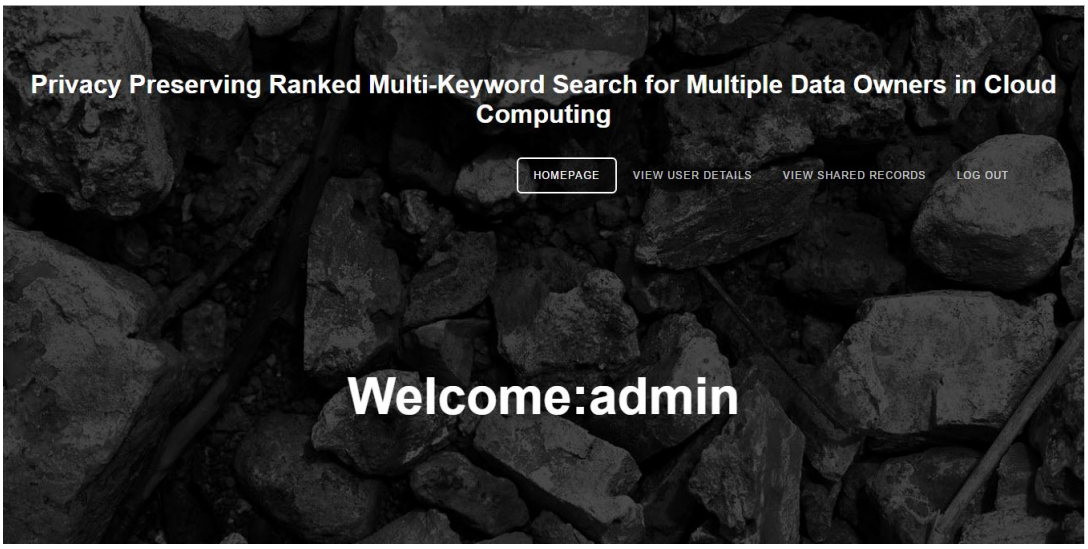


Figure 9.11: Admin Home Page

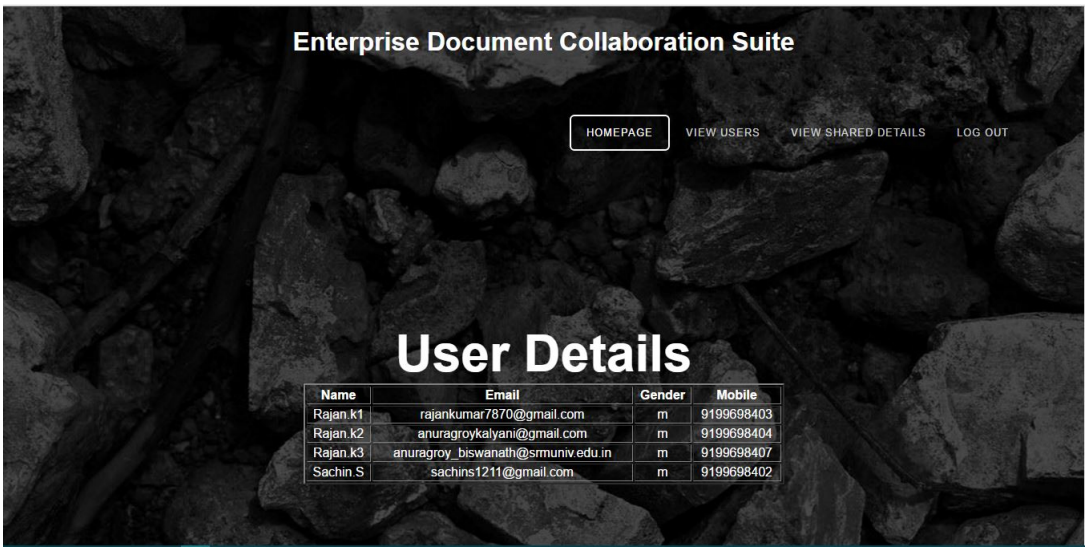


Figure 9.12: Admin View Users

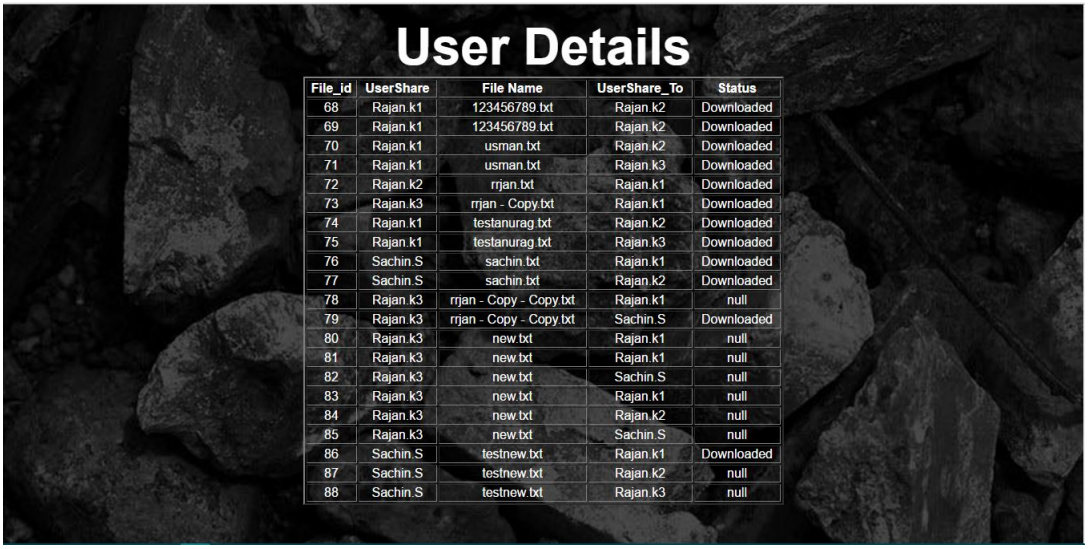


Figure 9.13: Admin View Shared File

CHAPTER 10

CONCLUSION

In this paper, we take care of the issue of secure multi keyword search for multiple data owners in the cloud computing.

We presented a secret key generation protocol and client authentication protocol which is utilized to shield the framework from assailants and verify just the enrolled clients.

CHAPTER 11

FUTURE ENHANCEMENT

In our future work, we work on the safe fuzzy keyword search in multiple data users and we are planning to implement on the commercial cloud.

REFERENCES

1. C. Bosch, R. B. and Hartel, P. (2011). "Conjunctive wildcard search over encrypted data." *IEEE Transactions on Parallel and Distributed Systems*, 114–127.

2. C. Wang, S. S. Chow, Q. W. K. R. and Lou, W. (2013). "Privacy preserving public auditing for secure cloud storage." *Journal of Vibration and Acoustics*, 8(2), 362âĂŞ375.

3. J. W. Li, J. Li, X. F. C. and et al (2012). "Efficient keyword search over encrypted data with fine-grained access control in hybrid cloud." *In: Network and System Security*, 24(6), 490– 502.

4. P. Xu, H. Jin, Q. W. and Wang, W. (2013). "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack." *Computers, IEEE Transactions on*, 62(11), 2266âĂŞ2277.

5. R. Curtmola, J. Garay, S. K. and Ostrovsky, R. (2006). "Searchable symmetric encryption: improved definitions and efficient constructions." *In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press*, 79–88.

6. Sun, W. and et al (2013). "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking." *Proceedings of ACM SIGSAC*.

7. T. Jung, X. Y. Li, Z. W. and Wan, M. (2013). "Privacy preserving cloud data access with multi-authorities." *Proc. IEEE INFOCOM13*, 8(2), 2625–2633.

8. W. Sun, B. Wang, N. C. M. L. W. L. Y. T. H. and Li, H. (2013). "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking." *in Proc. IEEE ASIACCS13*, 71âĂŞ81.

9. Wei Zhang, Yaping Lin, S. X. J. W. and Zhou, S. (2014). "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing." *Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 276âĂŞ286.

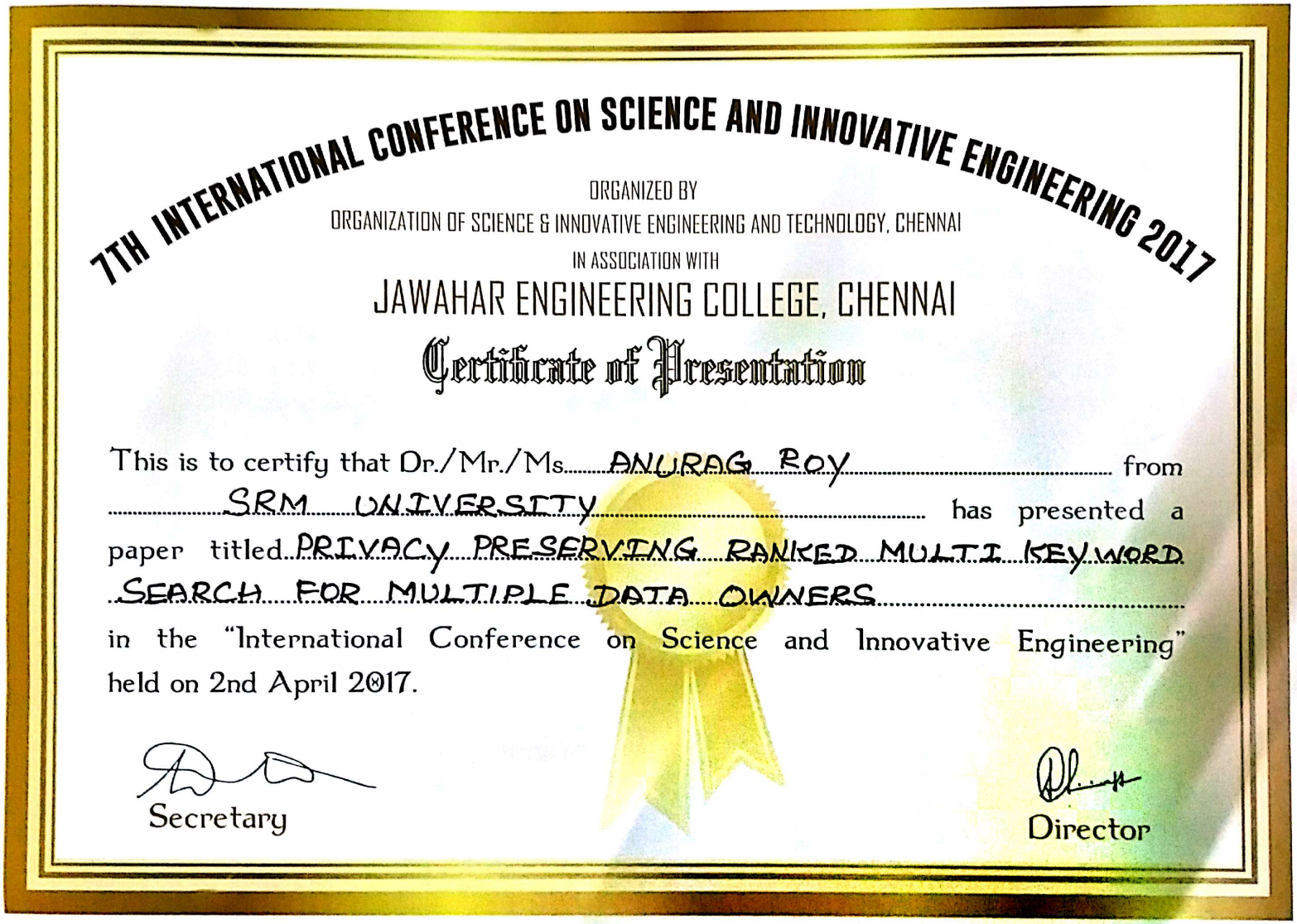
10. X. Liu, Y. Zhang, B. W. and Yan, J. (2013). "Secure multiowner data sharing for dynamic groups in the cloud." *IEEE Transactions on Parallel and Distributed Systems*, 24(6), 1182– 1191.

APPENDIX A

PRESENTATION

A.1 Certificates

Figure A.1: Rajan Kumar



7TH INTERNATIONAL CONFERENCE ON SCIENCE AND INNOVATIVE ENGINEERING 2017

ORGANIZED BY

ORGANIZATION OF SCIENCE & INNOVATIVE ENGINEERING AND TECHNOLOGY, CHENNAI

IN ASSOCIATION WITH

JAWAHAR ENGINEERING COLLEGE, CHENNAI

Certificate of Presentation

This is to certify that Dr./Mr./Ms. ANURAG ROY from SRM UNIVERSITY has presented a paper titled PRIVACY PRESERVING RANKED MULTI KEYWORD SEARCH FOR MULTIPLE DATA OWNERS in the "International Conference on Science and Innovative Engineering" held on 2nd April 2017.

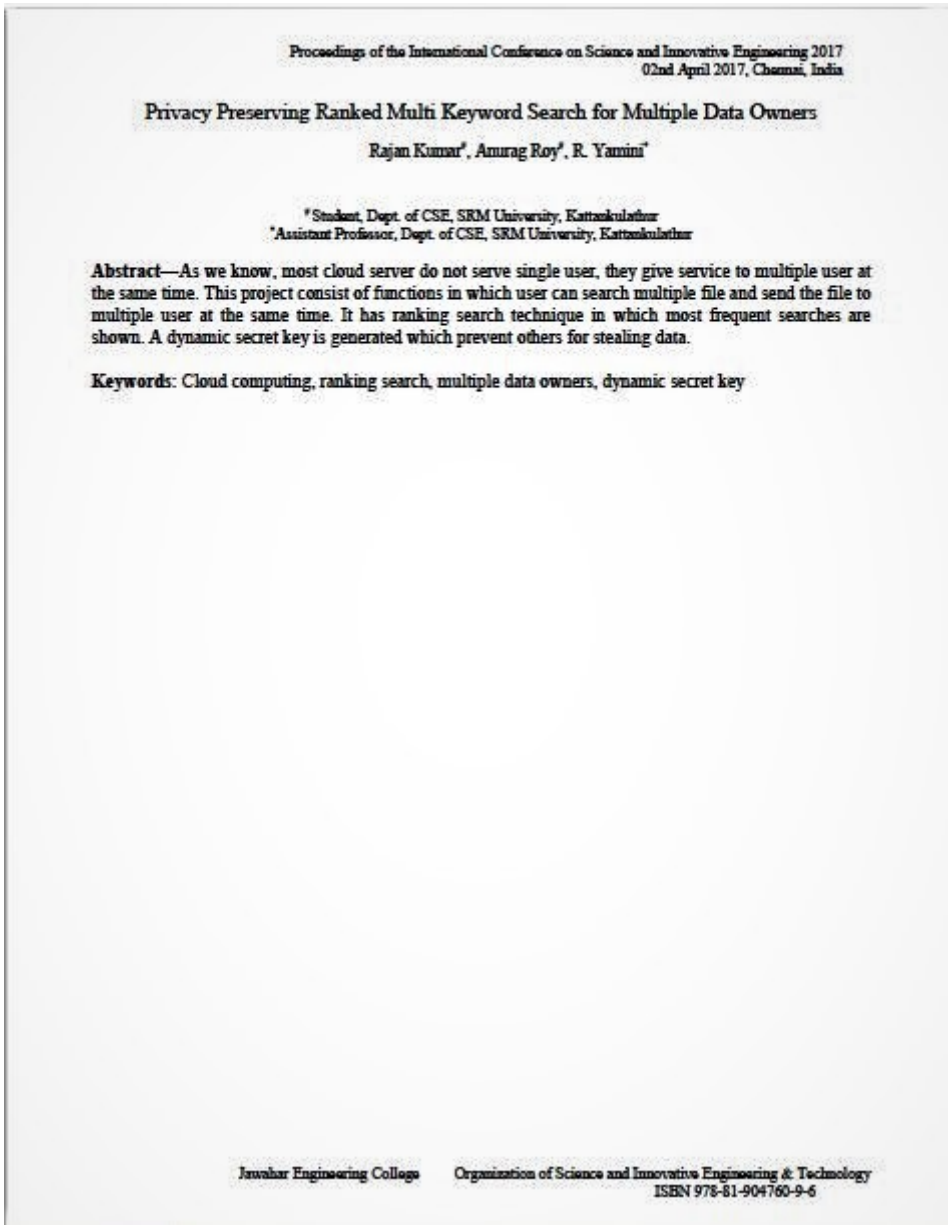
Secretary

Director

Figure A.2: Anurag Roy

APPENDIX B

CONFERENCE PROCEEDING ABSTRACT PAPER



Proceedings of the International Conference on Science and Innovative Engineering 2017
02nd April 2017, Chennai, India

Privacy Preserving Ranked Multi Keyword Search for Multiple Data Owners

Rajan Kumar*, Anurag Roy*, R. Yamini*

*Student, Dept. of CSE, SRM University, Kattankulathur
*Assistant Professor, Dept. of CSE, SRM University, Kattankulathur

Abstract—As we know, most cloud server do not serve single user, they give service to multiple user at the same time. This project consist of functions in which user can search multiple file and send the file to multiple user at the same time. It has ranking search technique in which most frequent searches are shown. A dynamic secret key is generated which prevent others for stealing data.

Keywords: Cloud computing, ranking search, multiple data owners, dynamic secret key

Jawahar Engineering College Organization of Science and Innovative Engineering & Technology
ISBN 978-81-904760-9-6

Figure B.1: Abstract Paper

APPENDIX C

PLAGIARISM REPORT

anurag.pdf

ORIGINALITY REPORT

1%	1%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijcta.com Internet Source	1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 20 WORDS
EXCLUDE BIBLIOGRAPHY	ON		

Figure C.1: Plagiarism Report