A

Major Project

On

CONTROL OF ACCESS TO ENCRYPTED CLOUD STORAGE

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

PINNINTI SAI PREETHI (197R1A05N9)

NANAM PREETHI (197R1A05N6)

AMARTHALURI CHANDRIKA (197R1A05J3)

Under the Guidance of

RAKSHITHA OKALI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, NewDelhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"CONTROL OF ACCESS TO ENCRYPTED CLOUD STORAGE"** being submitted by **PINNINTI SAI PREETHI (197R1A05N9), NANAM PREETHI (197R1A05N6) & AMARTHALURI CHANDRIKA (197R1A05J3)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Rakshitha Okali
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Rakshitha Okali**, Assistant Professor for her exemplary guidance, monitoring, and constant encouragement throughout the project work. The blessing, help, and advice given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. K. Shilpa, Dr. M . Subha Mastan Rao & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head of the Department of Computer Science and Engineering, **Dr. Ashuthosh Saxena**, Dean R&D, and **Dr. D T V Dharmajee Rao,** Dean Academics for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

PINNINTI SAI PREETHI (197R1A05N9)
NANAM PREETHI (197R1A05N6)
AMARTHALURI CHANDRIKA (197R1A05J3)

ABSTRACT

Cloud computing has become a popular way to store data, but many people are concerned about entrusting sensitive information to cloud providers who may not provide enough user control. To address this, many data owners choose to outsource encrypted data using Ciphertext-Policy Attribute-based Encryption (CP-ABE) for fine-grained access control. However, previous schemes have not adequately protected against Economic Denial of Sustainability (EDoS) attacks, where attackers can consume cloud resources and cost the payer a great deal of money. This lack of transparency and accountability is a major concern for data owners. In response, your proposal aims to secure encrypted cloud storage from EDoS attacks and provide resource consumption accountability by using CP-ABE schemes in a black-box manner that complies with the arbitrary access policies of CP-ABE. You have presented two protocols for different settings and conducted performance and security analyses to demonstrate the effectiveness and efficiency of the proposed solution.

Keywords: Ciphertext-policy attribute-based encryption(CP-ABE), Cloud Computing, Access Control, Protocols, Encryption.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Attribute-based encryption (ABE) is a type of encryption that allows for access control based on attributes like user characteristics or data properties. This is different from traditional access control methods, such as user names or roles. ABE can provide greater confidentiality compared to other methods, due to its ability to offer fine-grained control over who can access encrypted data.

ABE is highly flexible and efficient, making it applicable to various domains and applications. There are several types of ABE schemes, including key-policy ABE. ABE is making crucial advances in solving problems related to confidentiality, and its versatility and efficiency make it an attractive solution for many use cases.

1.2 PROJECT PURPOSE

The objective of this project is to find Attribute-Based Encryption (ABE) techniques that can guarantee confidentiality and allow for granular data access control in a cloud storage system. ABE has been shown to be highly effective in addressing various problems and offers great flexibility and efficiency, making it applicable to a wide range of applications such as cryptography, cloud computing, and more.

1.3 PROJECT FEATURES

The focus of this project is to implement Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to enable fine-grained and owner-centric access control for sharing encrypted files with other users. The project utilizes partially outsourced and fully outsourced protocols to ensure accountability for resource consumption.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System analysis is an important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

To ensure secure sharing of information, a general approach involves combining access control on the data owner side and on the cloud side. This involves encrypting the data and providing access to authorized users who can decrypt it using the access provided by the data owner. This approach enables granular access control and the ensures security of the shared information.

2.2 EXISTING SYSTEM

- In the existing system, R. K. Ko et.al., the authors discussed key issues and challenges about how to achieve accountability in cloud computing.
- In the literature, D. O´. Coilea´et.al.,the authors surveyed existing accounting and accountability in content distribution architectures.
- V. Sekaret.al. and C. Chen et. al., the authors respectively proposed a systematic approach for verifiable resource accounting in cloud computing.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- The accounting approach involves changes to the system model and requires the anonymous verification of users, which is not supported in previous systems.
- The access control is only available for data owners, which turns out to be insufficient.
- This makes the storage system vulnerable to resource-exhaustion attacks.
- It loses the flexibility of access control from CP-ABE.

2.3 PROPOSED SYSTEM

The security requirements of the system are achieved through two key components:

1. A cloud-side access control mechanism that blocks users whose attribute set does not meet the access policy A;
2. A proof-collecting subsystem that allows the cloud provider to collect proofs of resource consumption from users and present them to the data owner later.

Real-world scenarios often involve specifying a maximum expected download time, and data owners can remain offline unless they choose to increase this value. The first protocol, Partially Outsourced Protocol (POP), is designed to address such scenarios. In cases where the data owner cannot set expectations for download times or would be offline for an extended period, the Fully Outsourced Protocol (FOP) can be utilized, allowing the data owner to delegate control to the cloud.

Partially Outsourced Protocol (POP)

The Partially Outsourced Protocol (POP) involves encrypting an ephemeral key in CP-ABE by the data owner. This key is then used for both message encryption/decryption and cloud-side access control. The data owner provides the cloud provider with a set of N challenge ciphertexts $\{enchali\}\in[N]$ and the corresponding hashed challenges $\{hashi\}i\in[N]$.

To prove legitimacy to the cloud provider, the user must show that the decryption result $chalj$ of a randomly selected unused challenge ciphertext $enchalj$ is a pre-image of $hash$. If the user's response is valid, the cloud provider stores the response for further

resource consumption accounting.

To reduce storage space and improve efficiency, a bloom filter can be introduced for data owners to store their challenge plaintexts. This bloom filter can be stored locally or remotely on the cloud server. As the challenge update process cannot be outsourced to the cloud and must be implemented on demand or periodically by the data owner, the scheme is referred to as the Partially Outsourced Protocol (POP).

The procedure of POP is described in detail as follows:

1) Encrypt and Upload (POP-EU):

This operation is implemented by an individual data owner independently, which can be divided into the following four steps:

- ***POP-EU-1***: The data owner uses hybrid encryption to encrypt the message. The data owner randomly selects a symmetric key $\mathsf{k} \leftarrow_{\$} \{0,1\}^{\lambda}$ and uses the key to encrypt the message M. Then the data owner encrypts that symmetric key k with CP-ABE under $\mathbb{A}$:

$$\begin{aligned} \mathsf{c}_0 &\leftarrow \mathsf{AEAD.Enc}\,(\mathsf{k}, \textit{“message”} \parallel M)\,, \\ \mathsf{c}_1 &\leftarrow \mathsf{ABE.Enc}\,(\mathsf{mpk}, \mathsf{k}, \mathbb{A})\,, \\ \mathsf{c}_2 &\leftarrow \mathsf{SIG.Sign}\,(\mathsf{sk}_{\mathsf{owner}}, \mathsf{c}_1)\,. \end{aligned}$$

- ***POP-EU-2***: The data owner randomly generates N challenge plaintexts from the message space. They should be different with each other.

$$\{\mathsf{chal}_1, \mathsf{chal}_2, \ldots, \mathsf{chal}_N\},\ \mathsf{chal} \leftarrow_{\$} \{0,1\}^{L}.$$

The data owner generates the hashes of these challenges:

$$\mathsf{hash}_i = H(\mathsf{chal}_i), \forall i \in [1, N],$$

where $H(\cdot)$ is a collision-resistant hash function. For each challenge plaintext chal_i, the data owner uses k to encrypt it with a fixed prefix *“challenge”*. The prefix makes these challenges different from messages,

Now, we have

$$\begin{aligned} \mathsf{c}_3 &= \{\mathsf{hash}_i\}_{i\in[N]}, \\ \mathsf{c}_4 &= \{\mathsf{chal}_i\}_{i\in[N]}. \end{aligned}$$

- ***POP-EU-3***: The data owner creates a bloom filter to store the challenge plaintexts. We denote m as the size of the bloom filter.

$$\begin{aligned} &\mathsf{bf} \leftarrow \mathsf{BF.Setup}\,(m, \lambda)\,, \\ &\forall i \in [N],\ \mathsf{bf} \leftarrow \mathsf{BF.Insert}\,(\mathsf{bf}, \mathsf{chal}_i)\,. \end{aligned}$$

And then the data owner encrypts the bloom filter:

$$\mathsf{c}_5 = \mathsf{ABAE.Enc}\,(\mathsf{k}, \mathsf{bf})\,,$$

where k is the data owner's secret key. Note that to avoid the cloud provider understanding the structure of the bloom filter, the data owner should use its own **keyed** hash functions in the element insertion and test. We assume that the data owner keeps the version number of the bloom filter to thwart rollback attacks.

- ***POP-EU-4***: The following tuple is uploaded to the cloud:

$$\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4, \mathsf{c}_5)\,.$$

2) Cloud-side Access Control: POP-CR.

POP-CR-1: The cloud provider selects one of the unused challenge $enchal_j$ and sends the following tuple to the user:

- ***POP-CR-1***: The cloud provider selects one of the unused challenge enchal_j and sends the following tuple to the user:

$$(\mathsf{c}_1, \mathsf{c}_2, \mathsf{enchal}_j)\,.$$

The data user decrypts the ciphertexts and verifies the signature of the owner. The decryption of c_1 requires the data user to satisfy the policy $\mathbb{A}$:

$$\begin{aligned}&\text{HALT if } \mathsf{SIG.Verify}\,(\mathsf{vk}_{\mathsf{owner}}, \mathsf{c}_2, \mathsf{c}_1) = 0,\\ &\mathsf{k} \leftarrow \mathsf{ABE.Dec}\,(\mathsf{sk}_i, c_1)\,,\\ &\mathsf{chal}'_j \leftarrow \mathsf{AEAD.Dec}\,(\mathsf{k}, \mathsf{enchal}_j)\,.\end{aligned}$$

The data user sends chal'_j to the cloud provider.

- ***POP-CR-2***: The cloud checks $\mathsf{hash}_j \stackrel{?}{=} H(\mathsf{chal}'_j)$. If true, the cloud sends c_0 to the data user, which can be decrypted with the session key k and meanwhile the challenge as *used*. Otherwise, the cloud aborts. The user response chal'_j is the proof of the resource consumption accounting.

3) Challenge update (POP-SU):

The scheme allows for on-demand or periodic challenge updates by the data owner, as long as the specified upper bound of download times (N) has not yet been reached. If the data owner wishes to provide additional challenges, they can do so by being online for a short period. The update process is similar to that in the POP-EU-2 phase, using the same key k. The data owner is assumed to keep a record of session keys either in local storage or encrypted form outsourced to the cloud. As the plaintext space for challenges is sufficiently large, it is assumed that no duplicated challenge plaintexts are generated. If a bloom filter is used (and its encrypted form), it will need to be reconstructed in this case

4) Resource Accounting (POP-RA):

Data owners and the cloud interactively implement this operation.

sends back the encrypted bloom filter c_5 and m user responses $\{chal_i\}_{i=1,2,...,m}$. Given the probabilistic check rate β, $\beta \cdot m$ responses are randomly selected for verification:

$$(chal'_1, chal'_2, ..., chal'_{\beta m}) \leftarrow_{\$} (chal_1, chal_2, ..., chal_m).$$

The data owner decrypts the bloom filter c_5, only if integrity holds and the version number indicates the freshness, the data owner can accept the resource consumption if:

$$\sum_{i=1}^{\beta \cdot m} \mathsf{BF.Test}\left(bf, chal'_i\right) = \beta \cdot m.$$

Though the bloom filter has some false positives, it is sufficient to achieve the covert security against a cloud

Fully Outsourced Protocol (FOP)

The Fully Outsourced Protocol (FOP) is a protocol that allows for outsourced challenge generation and update, as well as resource accounting, without relying on an external PKI. It is based on the signature algorithm and offers two key differences when compared to the Proof of Possession (POP) protocol. Firstly, in FOP, the cloud provider generates the challenges {⟦enchal⟧i }(i∈[N]) instead of the data owners. Secondly, the data owners generate a pair of signature keys (vk, sk) for each file, which users can use to sign a confirmation to prove resource consumption.

The FOP procedure involves four steps:

- Encrypt and Upload (FOP-EU): The data owner encrypts the file using a symmetric key and uploads it to the cloud. The data owner also generates a pair of signature keys (vk, sk) for the file.
- Outsourced Challenge Generation (FOP-CG): The cloud provider generates the challenges {⟦enchal⟧i }(i∈[N]) for the file, which are then sent to the data owner. This step can be done in advance or on demand.
- Challenge-Response (FOP-CR): In this step, the data owners and the cloud run the operation. The data owner calculates a response to the challenges sent by the cloud and sends them back.
- Resource Accounting (FOP-RA): Legitimate users can sign a confirmation using the signature keys (vk,sk) to prove resource consumption. This operation is interactively implemented.

Overall, FOP provides a secure and efficient solution for outsourced challenges generation/update and resource accounting, without relying on an external PKI.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- We propose a general solution to secure encrypted cloud storage to prevent the EDoS attacks, as well as have fine-grained access control and resource consumption accountability.
- For different data owner online patterns and performance concern, we provide two protocols for authentication and resource consumption accounting.
- We also introduce the bloom filter and the probabilistic check to improve the efficiency but still guarantee the security.
- Compared with many state-of-arts constructions of encrypted cloud storage that assume the existence of a semi-honest cloud provider, we use a more practical threat model.
- Compared with relevant schemes, our approach works on the protocol level to provide the resource verifiability that relies on authorized users who satisfy the CP-ABE policy, and achieves the covert security which is more practical and secure.

2.4 FEASIBILITY STUDY

The feasibility of the project is being evaluated in this phase, and a business proposal will be presented outlining the general plan for the project along with estimated costs. To ensure that the proposed system is not a burden to the company, a feasibility study will be conducted during the system analysis. Understanding the major system requirements is crucial for conducting an effective feasibility analysis.

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

To ensure that the proposed system's development is financially feasible for the organization, a study was conducted to assess its potential impact on the company's finances. As with any project, the amount of funds allocated for research and development is limited, so it is important to justify all expenditures. Fortunately, most of the technologies used in the system are available free of charge, and only custom-made products needed to be purchased, which helped to keep the system development within the allocated budget..

2.4.2 TECHNICAL FEASIBILITY

The purpose of this study is to evaluate the technical feasibility of the proposed system by examining its technical requirements. It is essential that the system does not place an excessive burden on the available technical resources, as this could lead to issues like high costs and system failures. The developed system should have modest requirements to ensure that no or minimal modifications are necessary to implement it. This will facilitate the system's easy adoption by the organization without causing significant disruptions or necessitating major technical upgrades.

2.4.3 SOCIAL FEASIBILITY

Ensuring user acceptance of the system is an essential aspect that requires study. This involves providing the necessary training to enable users to use the system effectively without feeling intimidated. The level of user acceptance depends on the methods employed to educate and familiarize them with the system. The objective is to boost the user's confidence so that they can provide constructive feedback, which is valuable as they are the ultimate users of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces outline the logical attributes of each interface connecting the software product and the hardware components of the system. The subsequent requirements are some of the hardware requirements.

- **System** **:** Pentium IV 2.4 GHz.
- **Hard Disk** **:** 40 GB.
- **Floppy Drive** **:** 1.44 Mb.
- **Monitor** **:** 15 VGA Colour.
- **Mouse** **:** Logitech.
- **Ram** **:** 512 Mb.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system.
The following are some software requirements:

- **Operating System** **:** Windows 10
- **Technology** **:** Java 2 Standard Edition, JDBC
- **Web Server** **:** Tomcat 7.0
- **Editor** **:** NetBeans8.1

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

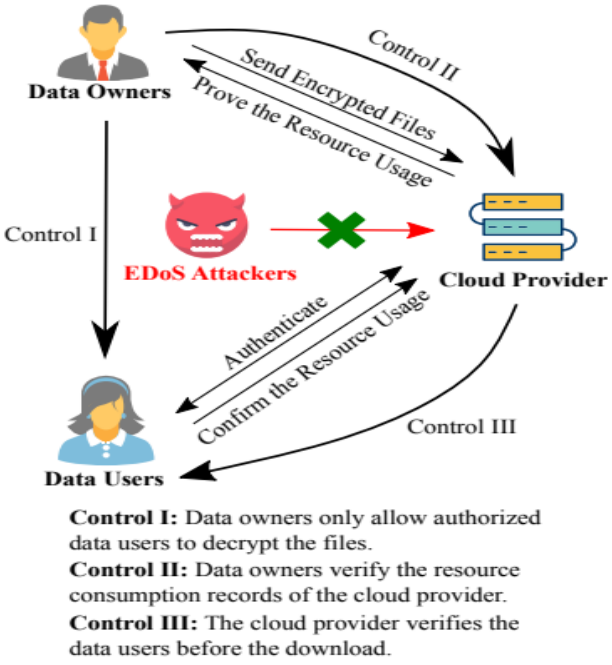


Figure 3.1 Project Architecture of Control of Access to Encrypted Cloud Storage

3.2 DESCRIPTION

The main objective of this project is to enhance the security of encrypted cloud storage by preventing attacks and ensuring accountability for resource consumption. The system will enable access to data only by authorized users, rather than making it available to everyone.

3.3 USE CASE DIAGRAM

A use case diagram is a graphical representation of the interactions between a user (or actors) and a system. It describes the functionality provided by the system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. Use case diagrams are an important tool for modeling the behavior of a system, as they allow designers to identify the different ways that users might interact with the system and to define the scope of the system's functionality.

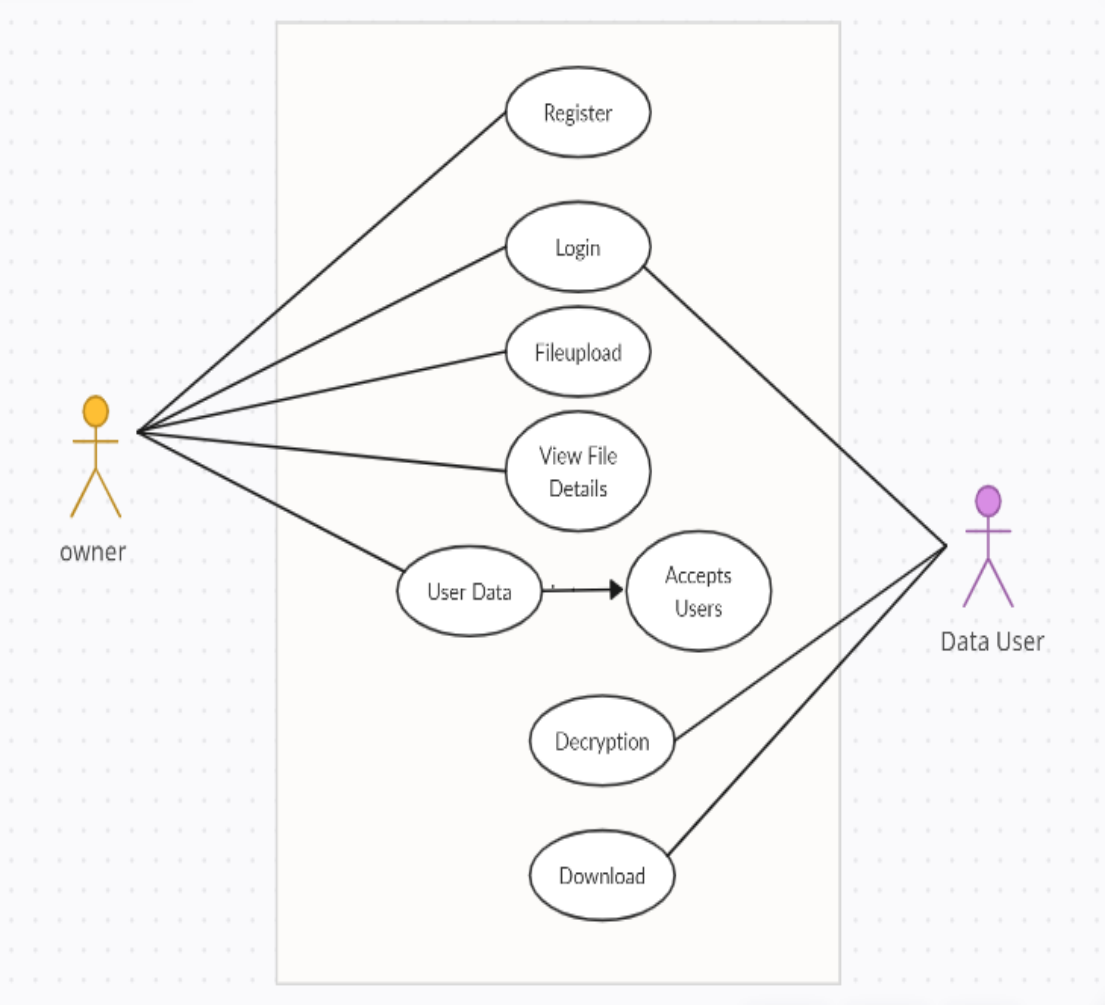


Figure 3.2 Use Case Diagram for Control of Access to Encrypted Cloud Storage

3.4 CLASS DIAGRAM

A class diagram is a static structure diagram in the Unified Modeling Language (UML) used in software engineering to describe the structure of a system. It provides a graphical representation of a system's classes, including their attributes and operations (methods), and the relationships between classes. Class diagrams show how classes are related to each other and which classes contain specific information.

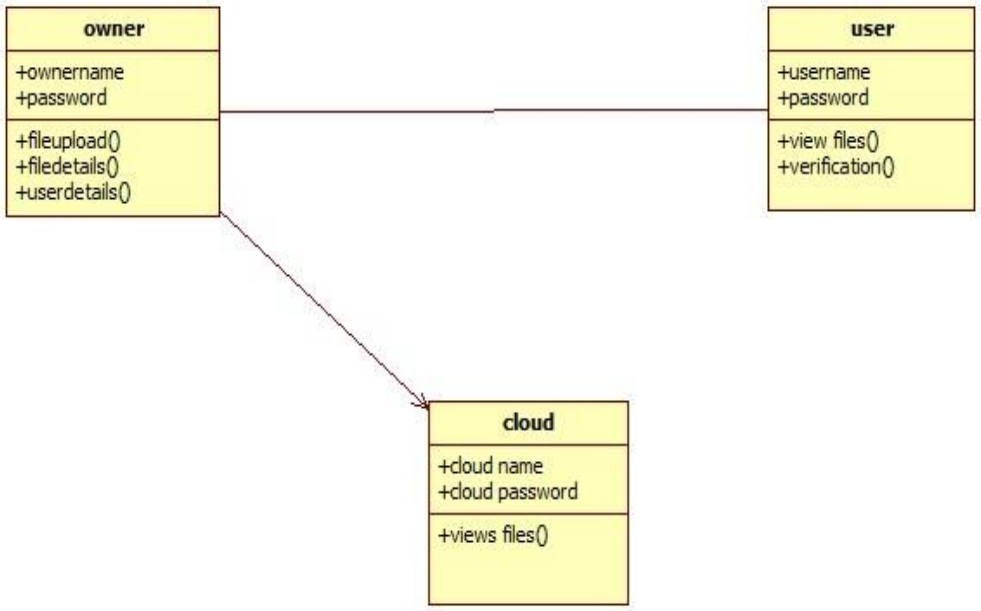


Figure 3.3 Class Diagram for Control of Access to Encrypted Cloud Storage

3.5 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

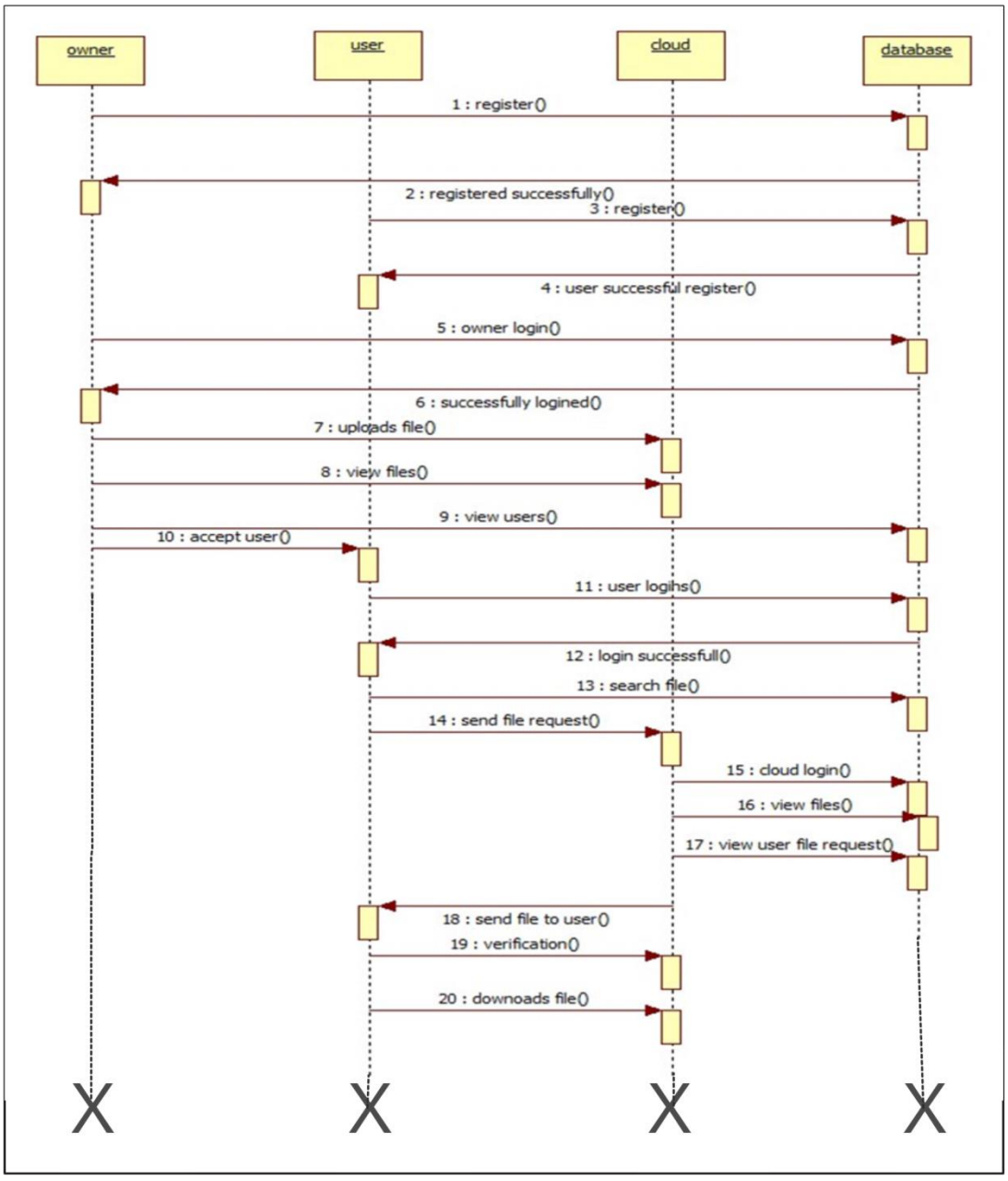


Figure 3.4 Sequence Diagram for Control of Access to Encrypted Cloud Storage

3.6 ACTIVITY DIAGRAM

Sequence diagrams are diagrams that show how objects interact with each other in a system in a sequential manner. They illustrate the order of interactions and events among the objects. These diagrams are also called event diagrams or event scenarios. Sequence diagrams are used to describe the functionality of objects in a system, and they are commonly used by software developers and business analysts to document and understand system requirements for both new and existing systems.

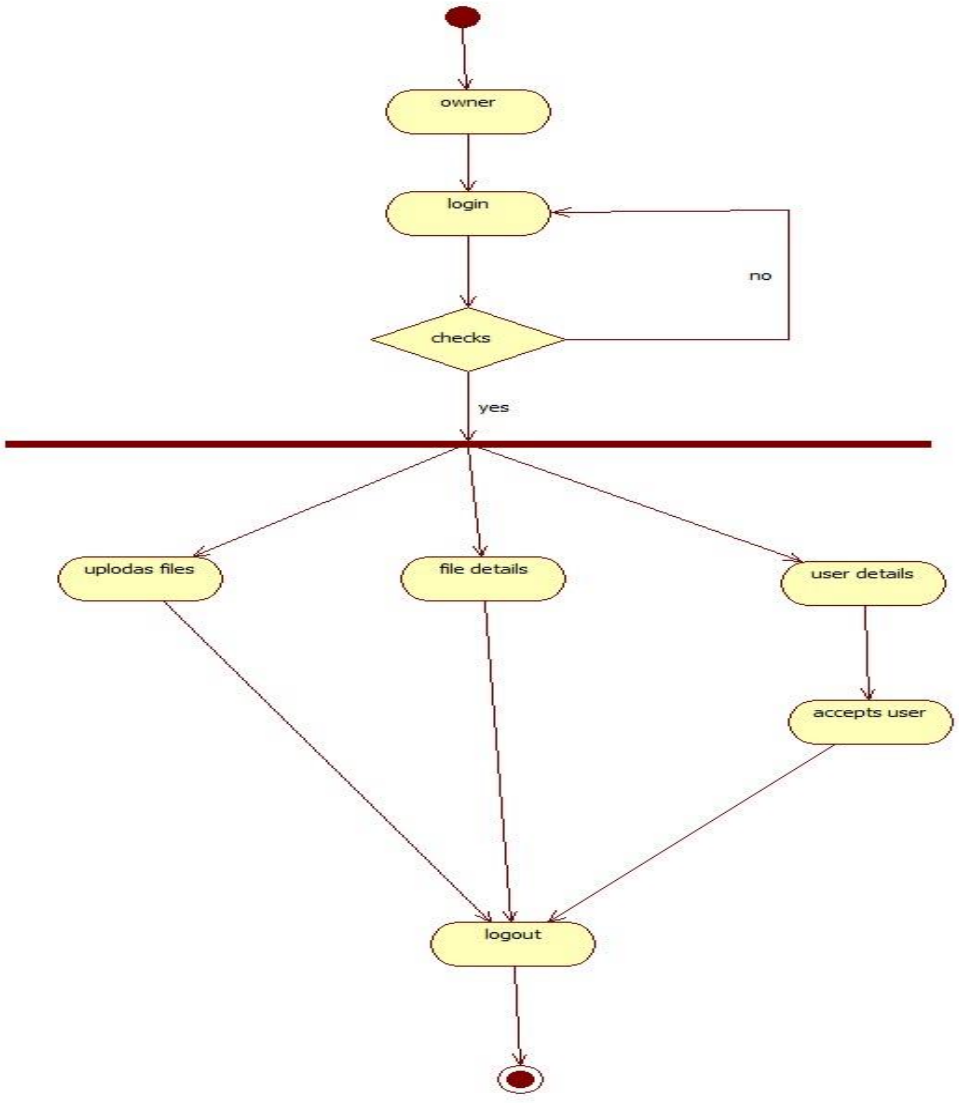


Figure 3.5 Activity Diagram for Control of Access to Encrypted Cloud Storage

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

Owner.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class Owner extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {
        response.setContentType("text/html");
        HttpSession session=request.getSession();
        PrintWriter out = response.getWriter();
        String user=request.getParameter("t1");
        String pass=request.getParameter("t2");
        try{
                String input[]={user,pass};
                String msg=DBConnection.ownerLogin(input);
                if(msg.equals("success")){
                        session.setAttribute("user",user);
                        RequestDispatcher
rd=request.getRequestDispatcher("OwnerScreen.jsp?t1=Welcome "+user);
                        rd.forward(request, response);
                } else {
                        response.sendRedirect("Owner.jsp?t1=Invalid User");
                }
```

```
}catch(Exception e){
                e.printStackTrace();
        }
    }
}
```

User.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class User extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)    throws
ServletException, IOException {
        response.setContentType("text/html");
        HttpSession session=request.getSession();
        PrintWriter out = response.getWriter();
        String user=request.getParameter("t1");
        String pass=request.getParameter("t2");
        try{
                String input[]={user,pass};
                String msg=DBConnection.userLogin(input);
                if(msg.equals("success")){
                        session.setAttribute("user",user);
                        RequestDispatcher
rd=request.getRequestDispatcher("UserScreen.jsp?t1=Welcome "+user);
                        rd.forward(request, response);
                } else {
                        response.sendRedirect("User.jsp?t1=Invalid User");
                }
```

```
        }catch(Exception e){
                e.printStackTrace();
        }
    }
}
```

Register.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.io.FileOutputStream;
import java.io.File;
public class Register extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)    throws
ServletException, IOException {
        response.setContentType("text/html");
        boolean flag=false;
        PrintWriter out = response.getWriter();
        String uname=request.getParameter("t1").trim();
        String pass=request.getParameter("t2").trim();
        String type=request.getParameter("t3").trim();
        String contact=request.getParameter("t4").trim();
        String secret=request.getParameter("t5").trim();
        String address=request.getParameter("t6").trim();
        try{
                BloomFilter.key = 50;
```

```
                String bloom_auth = BloomFilter.encrypt(secret);
                String input[]={uname,pass,type,contact,address,secret,bloom_auth};
                String res = DBConnection.createUser(input);
                if(res.equals("success")){
                        File userfile = new File(getServletContext().getRealPath("/")+"WEB-
INF/user/"+uname);
                        if(!userfile.mkdir())
                                userfile.mkdir();
                        RequestDispatcher
rd=request.getRequestDispatcher("Register.jsp?t1=You can login now. Bloom Filter From
Secret data : "+bloom_auth);
                        rd.forward(request, response);
                }else{
                        RequestDispatcher
rd=request.getRequestDispatcher("Register.jsp?t1=Error in registration process");
                        rd.forward(request, response);
                }
        }catch(Exception e){
                e.printStackTrace();
        }
}
}
```

Encrypt.java

```
package com;
import cpabe.Cpabe;
public class Encrypt{
public static boolean encrypt(String input,String enc,String policy){
        boolean flag = false;
        try{
                Cpabe att = new Cpabe();
                String public_key = input;
                att.enc(public_key,policy,enc,enc);
                flag = true;
        }catch(Exception e){
```

```
                e.printStackTrace();
                flag = false;}
        return flag;
}
}
```

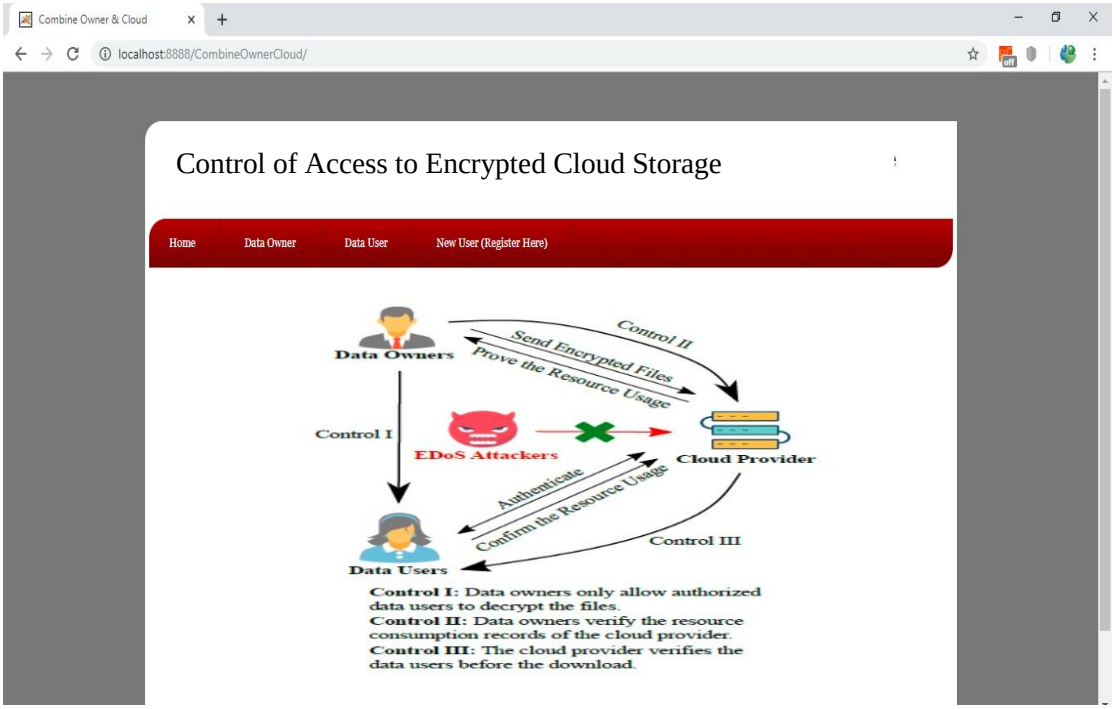
Decrypt.java

```
package com;
import cpabe.Cpabe;
import java.io.FileInputStream;
import java.io.File;
public class Decrypt {
        static String msg;
public static String getMsg(){
        return msg;
}
public static byte[] decrypt(String enc,String public_key,String private_key) {
        byte b[] = null;
        try{    File dec = new File(enc);
                dec = new File("E:/"+dec.getName());
                Cpabe test = new Cpabe();
                test.dec(public_key,private_key,enc,dec.getPath());
                FileInputStream fin = new FileInputStream(dec);
                b = new byte[fin.available()];
                fin.read(b,0,b.length);
                fin.close();
                dec.delete();
                msg = "success";
        }catch(Exception e){
                e.printStackTrace();
                b = "error".getBytes();
                msg = "error";}
        return b;
}
}
```

5. RESULTS

5. RESULTS

In below screen click on ‘New User’ link to add data owner or user. Here we are registering the data owner and the user to the cloud. So that the owner can share the files and the user can view and download whenever required.

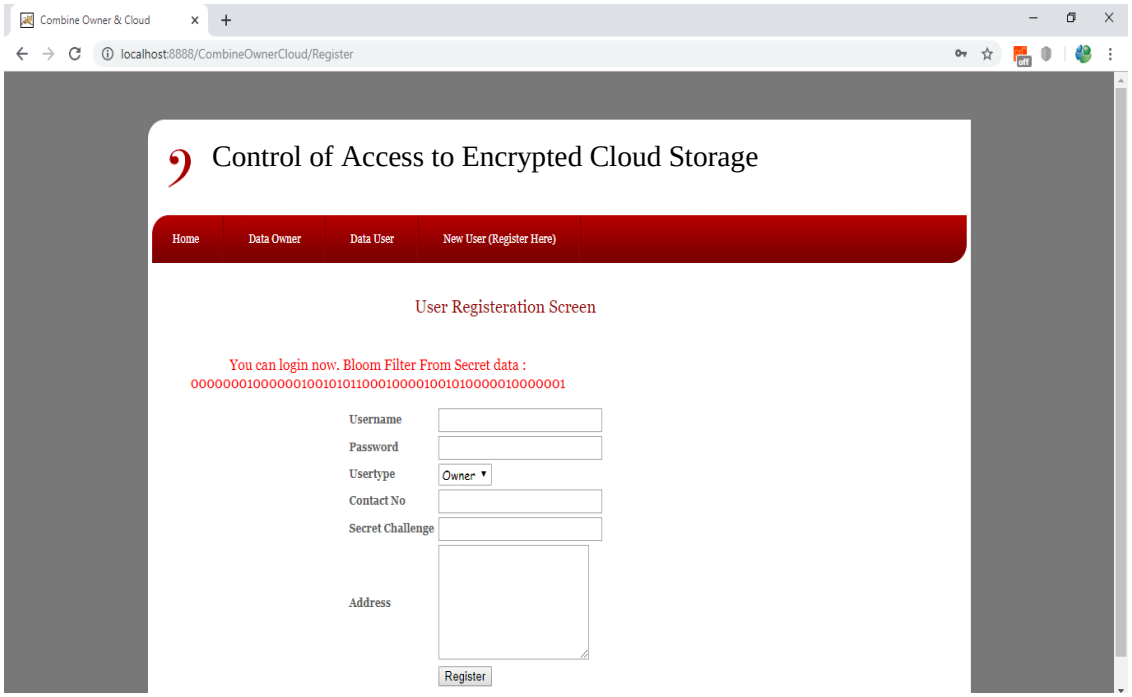


Screenshot 5.1: Home page to add data owner and user

In below screen I am adding one data owner and giving secret challenge as ‘kiran kumar’ and this secret data will be shared with all users of data owner. Without these secret data the users cannot After registration will get below screen.

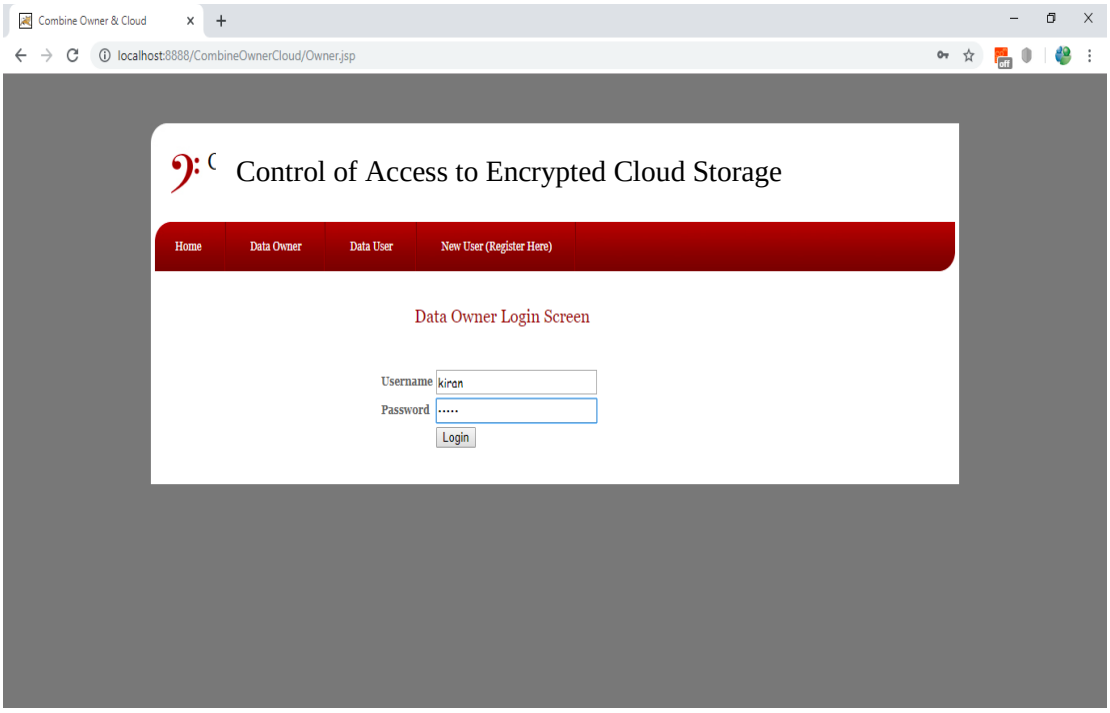
Screenshot 5.2: Giving secret challenge

In below screen for secret data application has generated BLOOM FILTER and this bloom filter will send to cloud to allow data user verification before download. Similarly you can add data users also. Now data owner can login and share data.



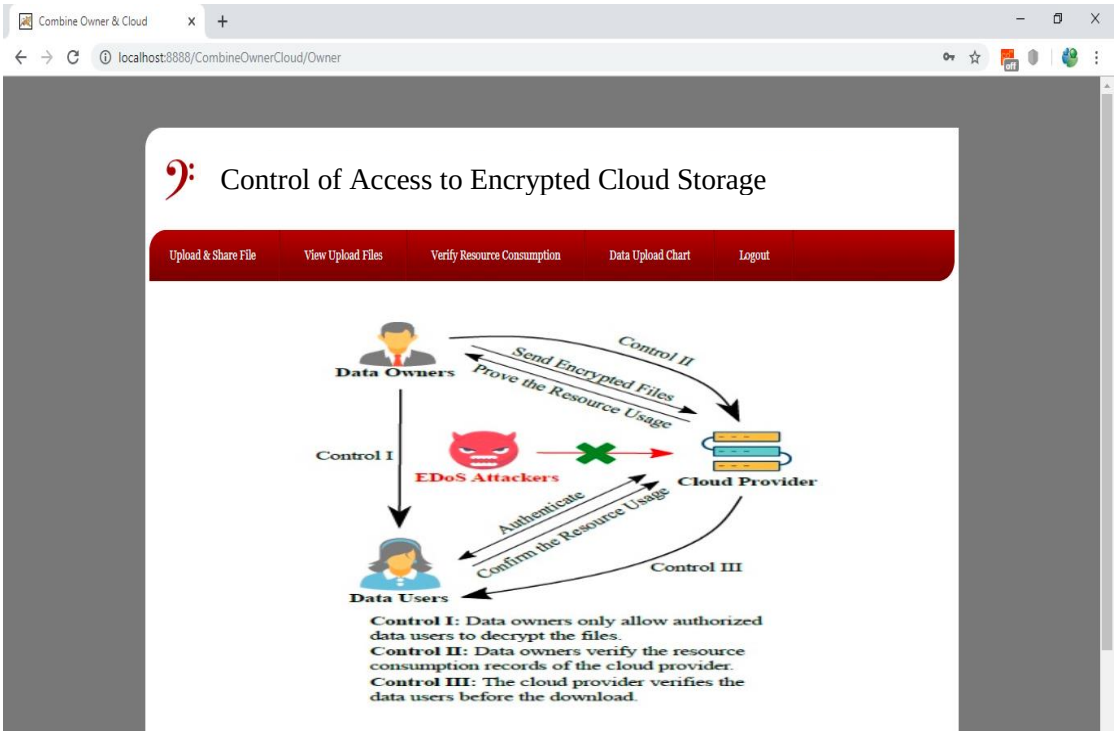
Screenshot 5.3: Generated Bloom filter

In below screen data owner login and after login will get below screen. Here the data owner will login to the cloud with the username and the password given at the time of registering the data owner.



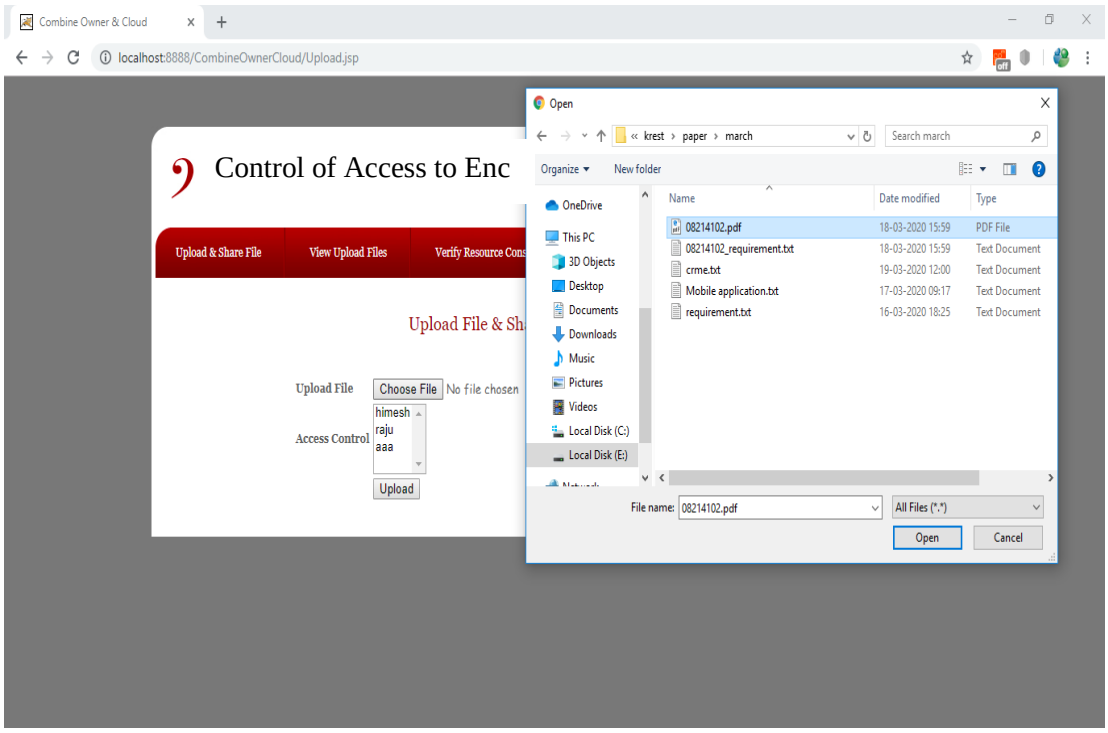
Screenshot 5.4: Data owner login

In below screen click on ‘Upload & Share File’ link to upload files. Here the data owner will share the files the users. So that the data users can view and download them whenever the user is required.



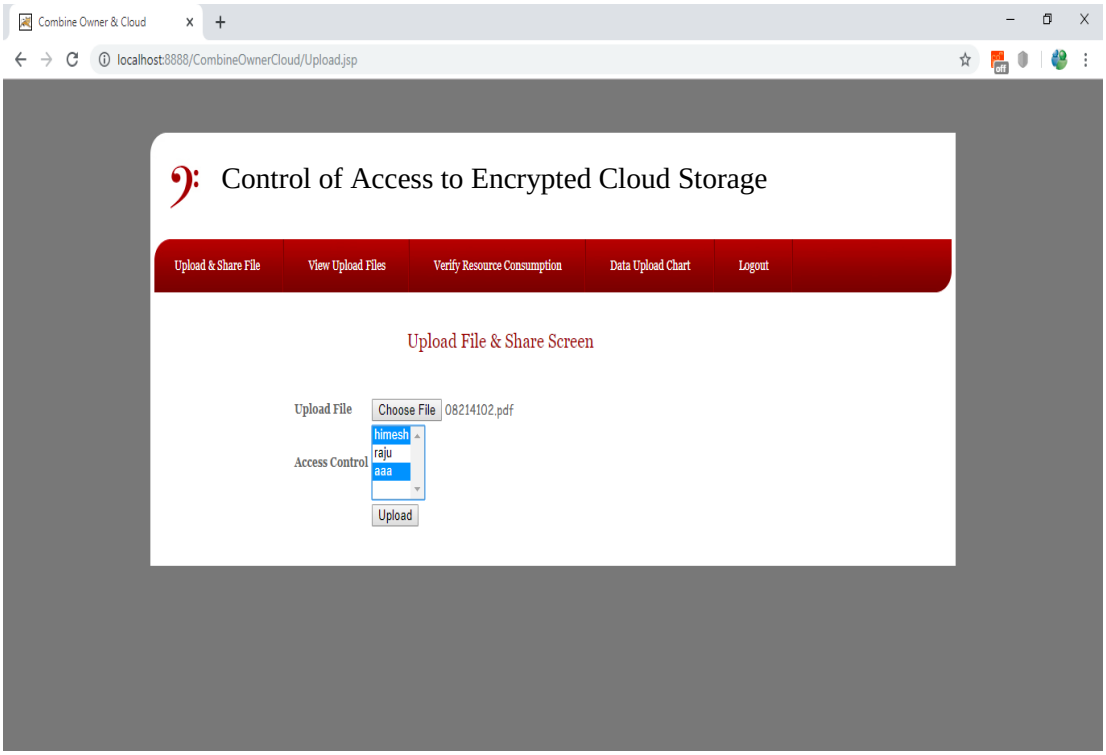
Screenshot 5.5: Upload file page

In below screen I am uploading one pdf file and then select share users from list by holding CTRL key. Here the data owner can upload any of the files to share with the data user and give access to view them.



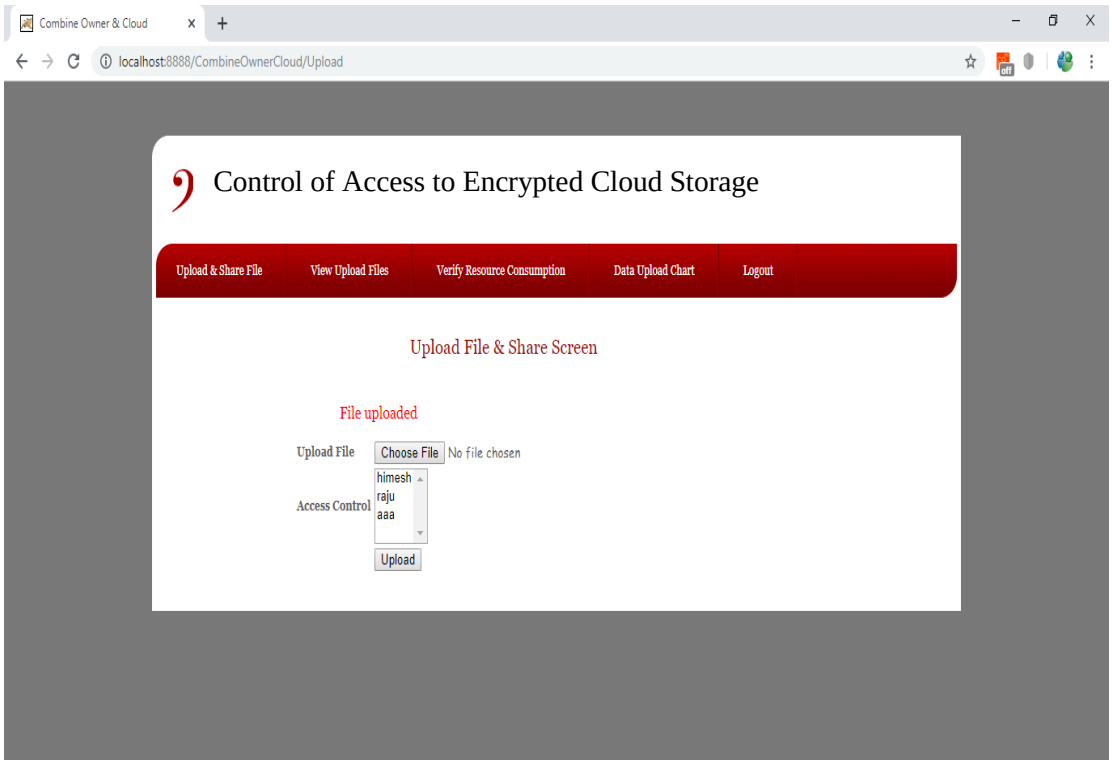
Screenshot 5.6 : Uploading the files

In below screen data owner giving permission to two users. Here the data owner has to select the users for viewing those uploaded files in the cloud and access them. Now click on ‘Upload’ button to upload file to cloud.



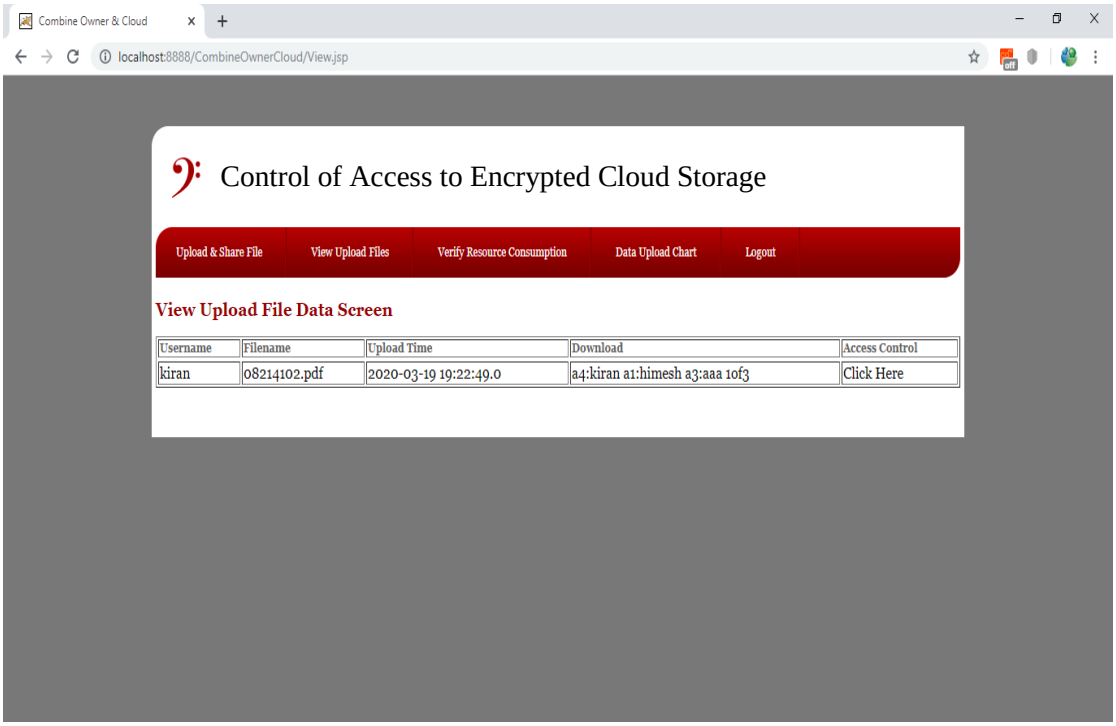
Screenshot 5.7: Uploading files to cloud page

In below screen we can see message as ‘File uploaded’ and now data owner can click on ‘View Upload File’ link to view all files uploaded by him. In this screen we can see that the files are uploaded by the data owner and given access to the authorized users.



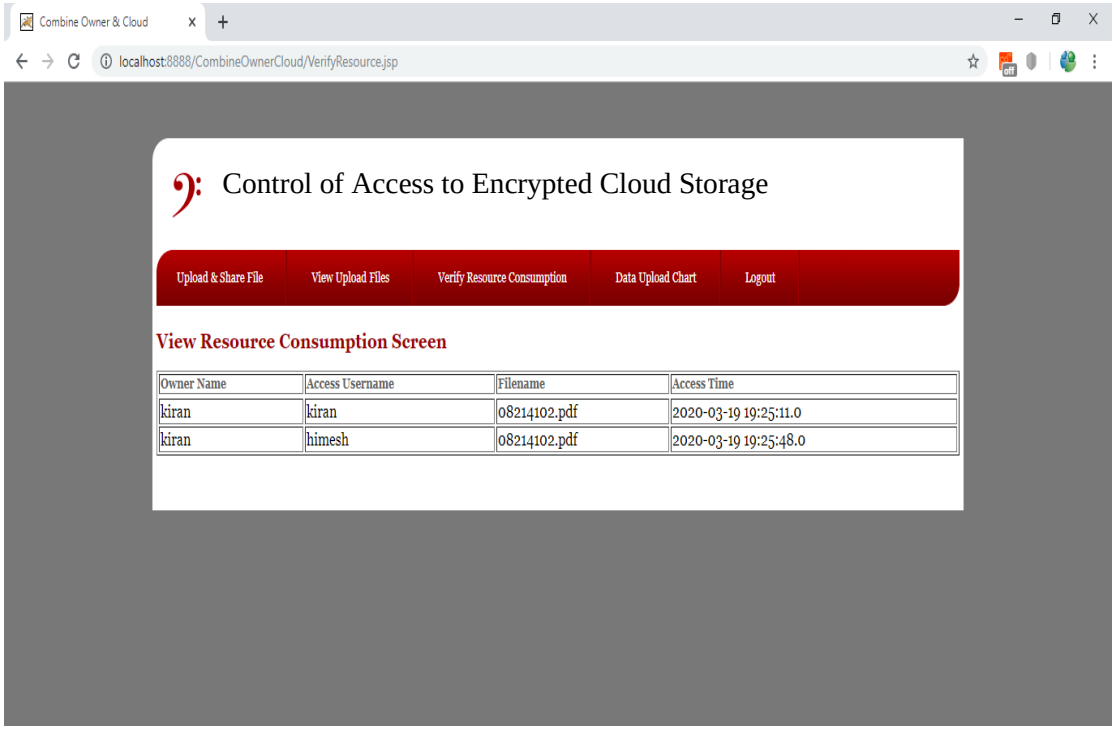
Screenshot 5.8: View uploaded files

In below screen data owner can download file by clicking on ‘Click Here’ link. All uploaded files store in encrypted format inside “WEB-INF/user” folder. In above screen we can see ACCESS POLICY also generated by data owner. Now click on ‘View Resource Consumption’ link to request cloud to give proof on resource consumption which means how many users access this data owner’s file. By using this option we can prevent cloud from cheating or applying fraud resource consumption cost.



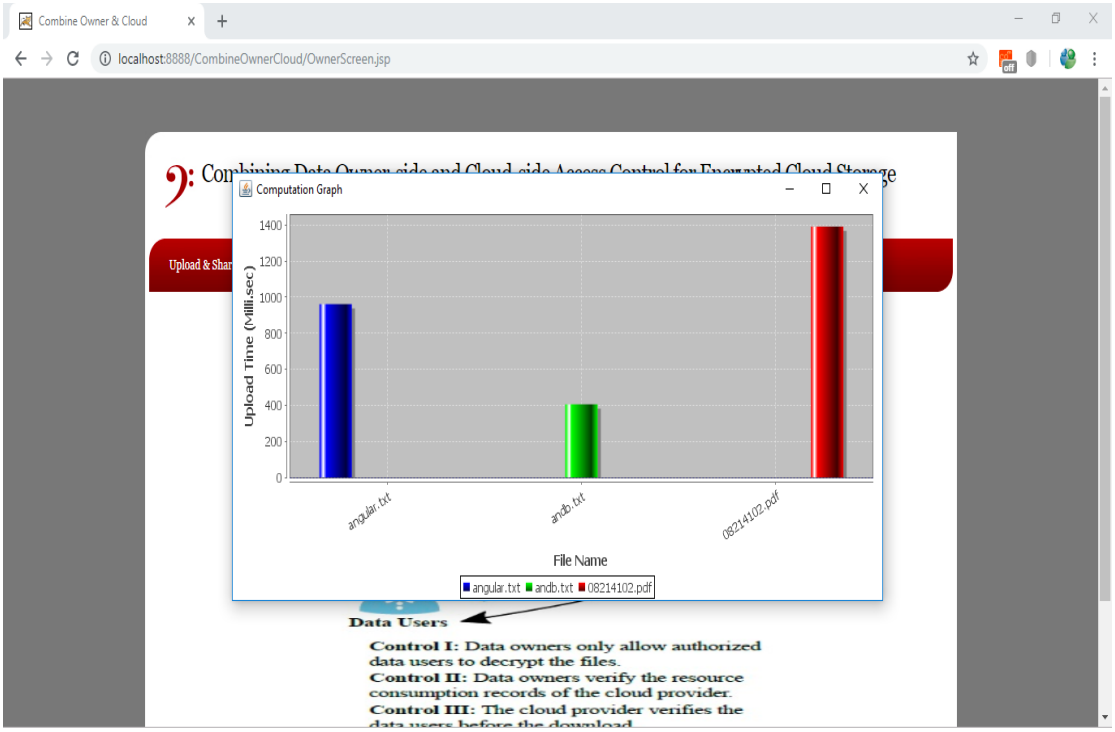
Screenshot 5.9: Data file screen

In below screen we can see list of all users who access this data owner's file. These all are the users who have access to the files shared by the data owner. Now click on 'Data Upload Chart' link to get data uploading execution time for each file



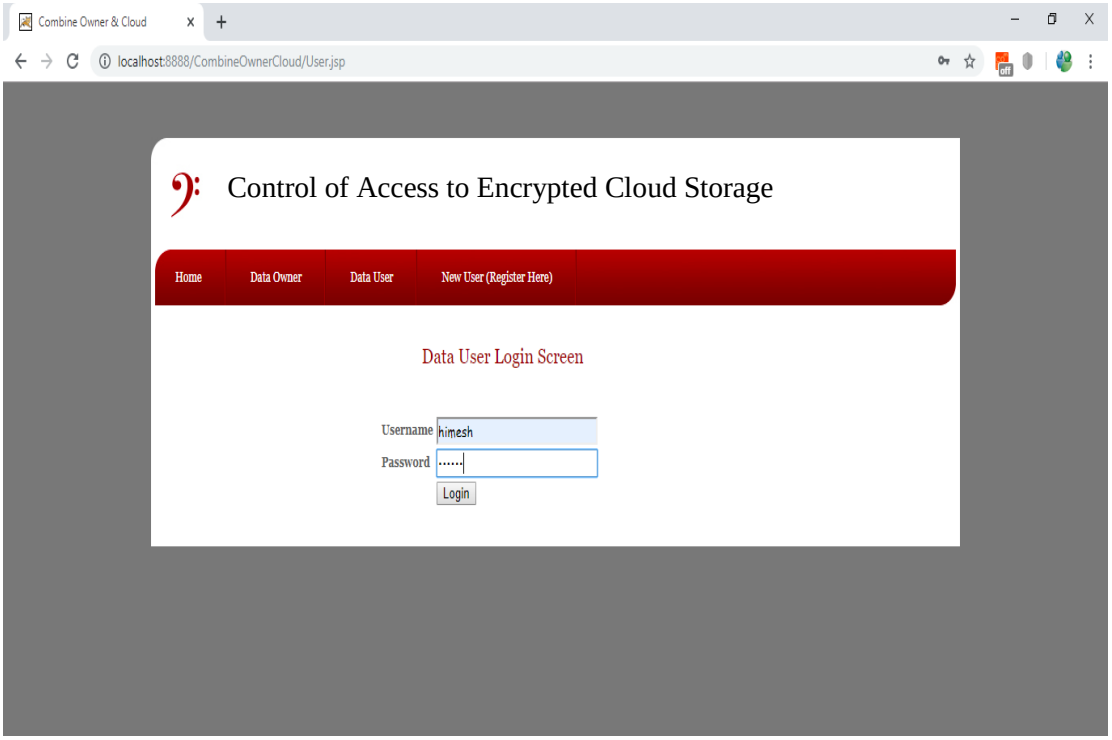
Screenshot 5.10: List of all users

In below graph x-axis represents file name and y-axis represents execution time to encrypt and upload that file to cloud. Now logout as the data owner and login as data user so that we can download file from the cloud.



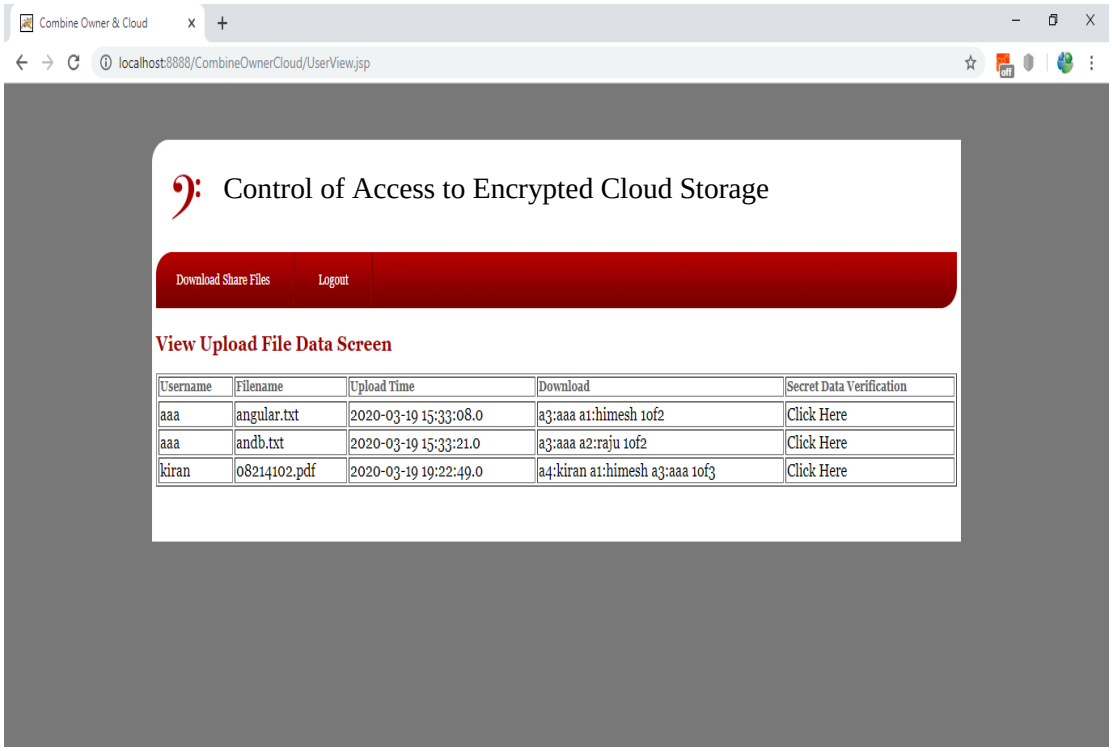
Screenshot 5.11: Data uploaded chart

In below screen data user is getting logged in with the username and password given at the time of registration. After login will get below screen which shows the user has logged to his account by providing the proper username and the password.



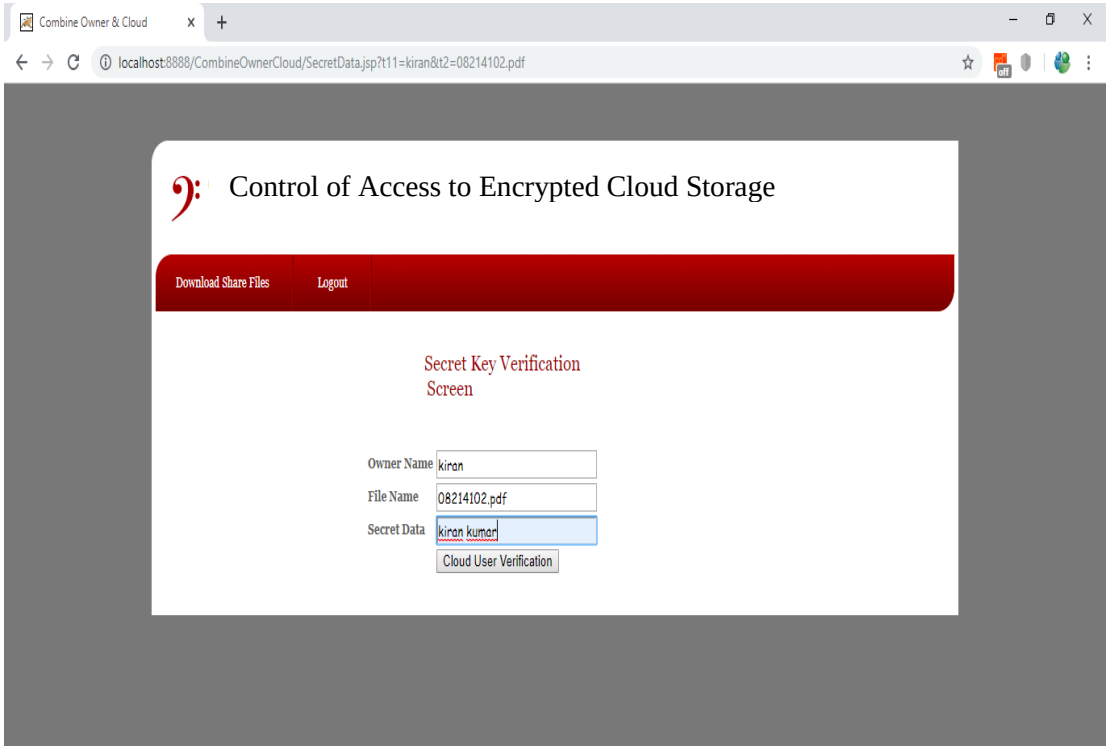
Screenshot 5.12: Data user login

In below screen data user can see all files from all data owners shared with the data users. The user can view and download By clicking on ‘Click Here’ link user can request cloud for file download and will get below screen.



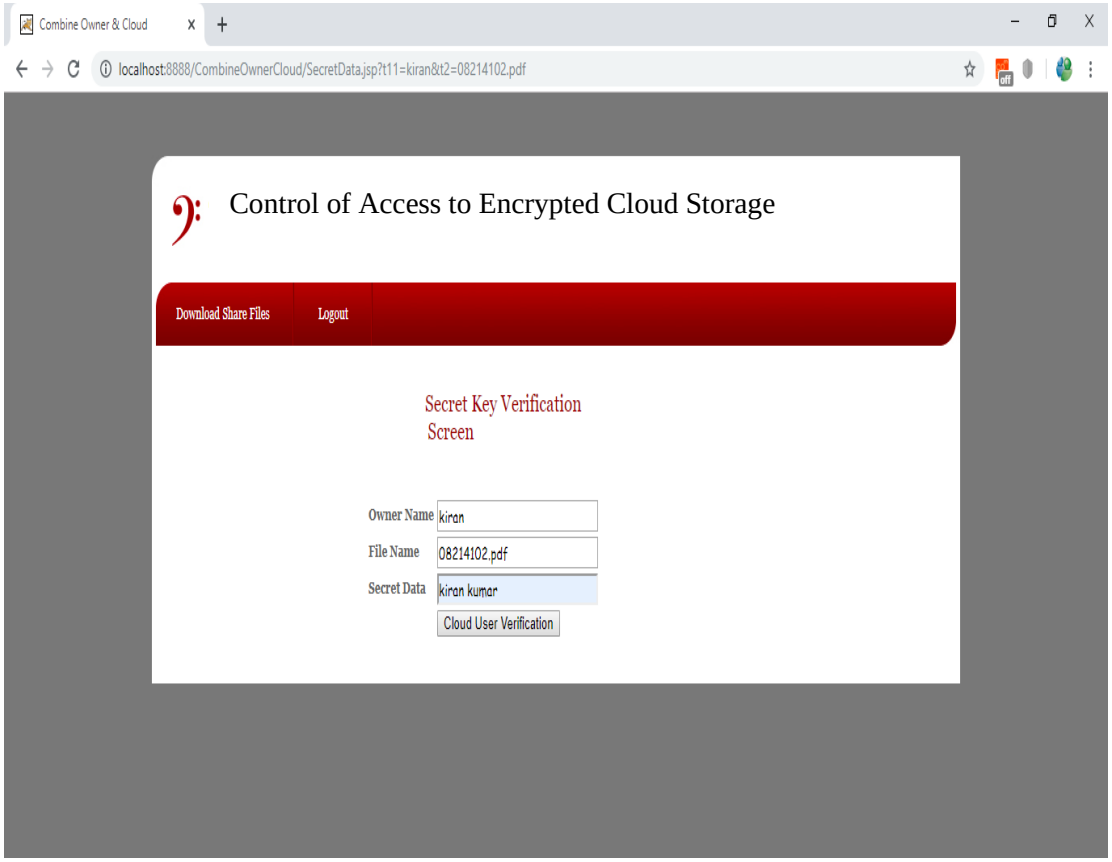
Screenshot 5.13: User requesting for file

In below screen cloud is asking data user for secret data challenge .If the data user provides the secret challenge given by the data owner correctly then only the data user can download the files.



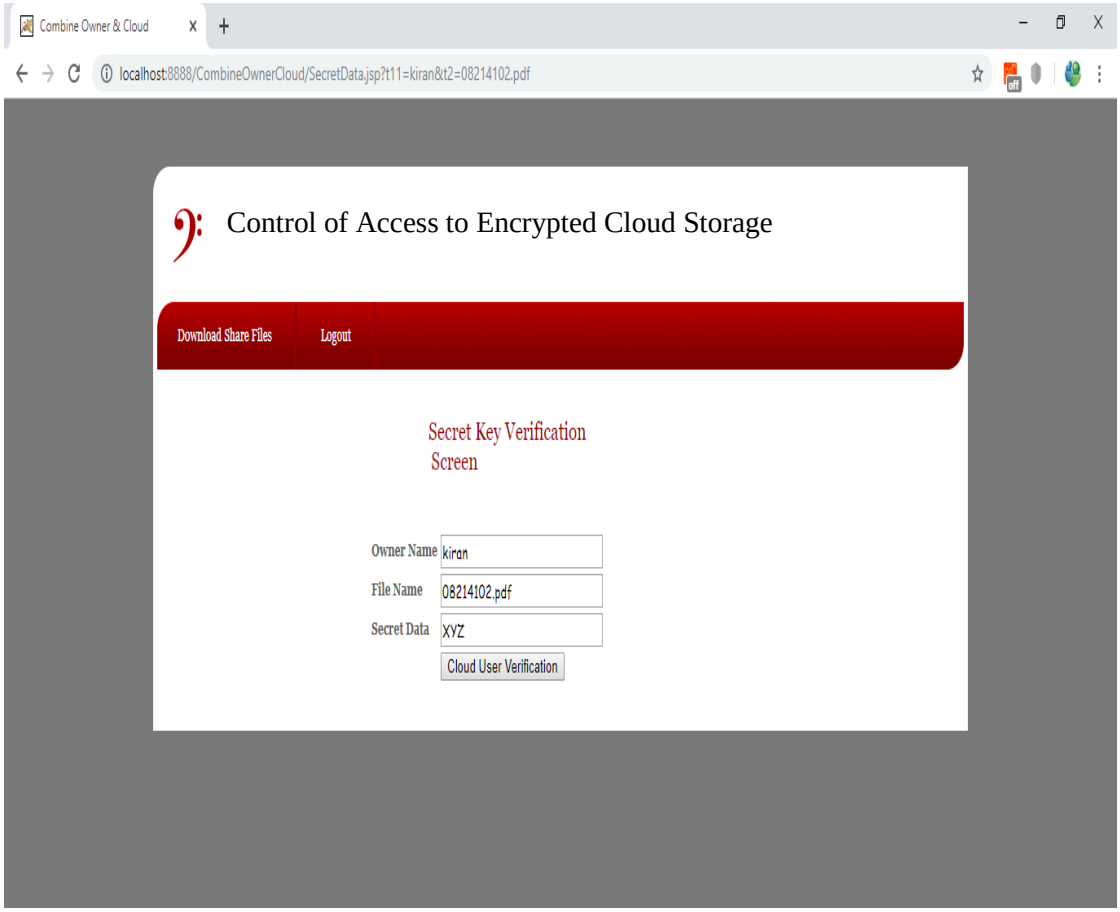
Screenshot 5.14: Cloud will ask the secret challenge

In below screen I am giving correct secret data and we can see file downloaded in browser status bar . We can get the access if the secret data is provided correctly otherwise we cannot access to the files shared by the user.



Screenshot 5.15: Giving secret data

After giving wrong secret challenge will get below screen. This is how it happens whenever we are providing the incorrect secret data at the time of accessing the files from the cloud.



Screenshot 5.16: If wrong data is entered

In below screen we can see secret data verification failed at cloud side and cloud will not allow user to download file. As it happened because the data user has provided the incorrect secret data.

Screenshot 5.17: Data verification failed

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The primary purpose of software testing is to verify that a system fulfills its requirements and meets user expectations, while identifying and resolving any errors or faults. This involves the use of various testing techniques to examine software functionality, which can be carried out on components, sub-assemblies, assemblies, or the entire system.

There are different types of software tests, each tailored to meet specific testing requirements. Testers can improve the quality and reliability of a software system by selecting and applying the appropriate test types.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing is a type of software testing where test cases are designed to validate the internal program logic and ensure that program inputs produce valid outputs. All decision branches and internal code flow are validated during unit testing. This testing is typically performed on individual software units of an application after their development but before they are integrated with other units.

Unit testing is a type of structural testing that requires knowledge of the system's construction and is considered invasive. It involves performing basic tests at the component level to test a specific business process, application, or system configuration. The primary objective of unit testing is to ensure that each unique path of a business process performs accurately according to documented specifications and contains clearly defined inputs and expected results. By performing unit testing, the quality and reliability of software systems can be improved.

6.2.2 INTEGRATION TESTING

Integration testing is a software testing technique that verifies the behavior of integrated software components to ensure they function as a cohesive program. Unlike

unit testing, which is focused on individual components, integration testing examines the interaction between these components and evaluates their combined functionality. This testing approach aims to identify any issues that may arise from the integration of components that may have been independently tested successfully. Integration testing is therefore essential in ensuring that a software system performs as expected and delivers the desired outcomes.

6.2.3 FUNCTIONAL TESTING

Functional tests are conducted to demonstrate that software functions are available and operate in accordance with the technical and business requirements, system documentation, and user manuals. These tests aim to ensure that each function performs as expected and produces the desired results, including inputs, outputs, and error handling. By verifying the functionality of the software, functional testing can help improve software quality and reliability, as well as increase user satisfaction.
Functional testing is centred on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test case. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	File	To access the file in secured manner	The sender gives the input in the form of a file with secret key	An output accessed by user with a secret key to open a file
2	File	To access the file in secured manner	The sender gives the input in the form of a file with secret key	The output accessed by user 2 with a secret key to open a file

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Our proposed project aims to enhance the security of encrypted cloud storage by combining access control measures both at the cloud and data owner sides. This approach is designed to resist Distributed Denial of Service (DDoS) and Extreme Denial of Service (EDoS) attacks while also providing resource consumption accounting. Our system can support arbitrary Conjunctional Policy Attribute-Based Encryption (CP-ABE) constructions, which are secure against malicious data users and covert cloud providers. We have relaxed the security requirements for the cloud provider to accommodate covert adversaries, a more practical and relaxed notion compared to semi-honest adversaries. To optimize resource consumption accounting and reduce overhead, we utilize bloom filters and probabilistic checks. Based on our performance analysis, our approach offers small overhead compared to existing systems.

7.2 FUTURE SCOPE

In this research work, we aim to develop a novel approach for ensuring secure file storage in the cloud, while preventing unauthorized access by other users. The proposed method involves encrypting the files and storing them in a secure cloud-based environment. This ensures that only authorized users are able to access the files and eliminates the risk of data breaches or unauthorized data access. The approach is expected to have significant benefits for organizations and individuals who store sensitive data in the cloud, by providing a secure and efficient means of managing their data.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in 2007 IEEE Symposium on Security and Privacy (SP’07). IEEE, 2007, pp. 321–334.

[2] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Advances in Cryptology–EUROCRYPT 2005. Springer, 2005, pp. 457–473.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proceedings of the 13th ACM conference on Computer and communications security (CCS2006). ACM, 2006, pp. 89–98.

[4] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in The 29th IEEE International Conference on Computer Communications (IEEE INFOCOM 2010). IEEE, 2010, pp. 1–9.

[5] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in Public Key Cryptography–PKC 2011. Springer, 2011, pp. 53–70.

[6] W. Li, K. Xue, Y. Xue, and J. Hong, “TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage,” IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, pp. 1484–1496, 2016.

[7] T. V. X. Phuong, G. Yang, and W. Susilo, “Hidden ciphertext policy attribute-based encryption under standard assumptions,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 35–45, 2016.

[8] K. Yang, X. Jia, and K. Ren, “DAC-MACS: Effective data access control for multi-authority cloud storage systems,” in Proceedings of the 32nd IEEE International Conference on Computer Communications (Infocom2013). IEEE, 2013, pp. 2895–2903.

[9] K. Xue, Y. Xue, J. Hong, W. Li, H. Yue, D. S. Wei, and P. Hong, “RAAC: Robust and auditable access control with multiple attribute authorities for

public cloud storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 953–967, 2017.

[10] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Computing Surveys, vol. 39, no. 1, p. 3, 2007.

[11] J. Idziorek and M. Tannian, "Exploiting cloud utility models for profit and ruin," in Proceedings of 2011 IEEE International Conference on Cloud Computing (CLOUD2011). IEEE, 2011, pp. 33–40.

[12] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245–2254, 2014.

[13] Q. Chen, W. Lin, W. Dou, and S. Yu, "CBF: a packet filtering method for DDoS attack defense in cloud environment," in Proceedings of IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC2011). IEEE, 2011, pp. 427–434.

[14] C. Hoff, "Cloud computing security: From DDoS (distributed denial of service) to EDoS (economic denial of sustainability)," http://www. rationalsurvivability.com/blog/?p=66.

[15] M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-Shield - a twosteps mitigation technique against edos attacks in cloud computing," in Proceedings of 4th IEEE International Conference on Utility and Cloud Computing (UCC2011). IEEE, 2011, pp. 49–56.

[16] J. Idziorek, M. Tannian, and D. Jacobson, "Attribution of fraudulent resource consumption in the cloud," in Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD2012). IEEE, 2012, pp. 99–106.

[17] R. K. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "TrustCloud: A framework for accountability and trust in cloud computing," in 2011 IEEE World Congress on Services (SERVICES 2011). IEEE, 2011, pp. 584–588.

[18] D. O. Coile ´ ain and D. O'mahony, "Accounting and accountability in ´ content distribution architectures: A survey," ACM Computing Surveys (CSUR), vol. 47, no. 4, p. 59, 2015.

[19] V. Sekar and P. Maniatis, “Verifiable resource accounting for cloud computing services,” in Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM, 2011, pp. 21–26.

[20] C. Chen, P. Maniatis, A. Perrig, A. Vasudevan, and V. Sekar, “Towards verifiable resource accounting for outsourced computation,” in ACM SIGPLAN Notices, vol. 48, no. 7. ACM, 2013, pp. 167–178.

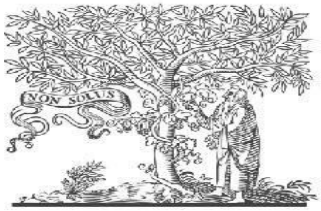
8.2 GITHUB LINK

https://github.com/Student-639/MAJOR-PROJECT/upload

9. PAPER PUBLICATION

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 29th Mar 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 03

10.48047/IJIEMR/V12/ISSUE 03/52

Title **FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE USING CP-ABE ENCRYPTION TECHNIQUE**

Volume 12, ISSUE 03, Pages: 365-373

Paper Authors
PINNINTI SAI PREETHI, NANAM PREETHI, AMARTHALURI CHANDRIKA,RAKSHITHA OKALI

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE USING CP-ABE ENCRYPTION TECHNIQUE

[1]PINNINTI SAI PREETHI, [2]NANAM PREETHI, [3]AMARTHALURI CHANDRIKA, [4]RAKSHITHA OKALI

[1,2,3]B. Tech Students, Dept. of CSE, CMR Technical Campus, Medchal, Hyderabad, Telangana, India.

[1]pinnintisaipreethi@gmail.com, [2]nanampreethi@gmail.com, [3]amarthaluri.chandrika@gmail.com

[4]Assistant Professor, Dept. of CSE, CMR Technical Campus, Medchal, Hyderabad, Telangana, India.

[4]rakshitaokali1997@gmail.com

ABSTRACT: Cloud computing has become a popular way to store data, but many people are concerned about entrusting sensitive information to cloud providers who may not provide enough user control. With the development of cloud storage system and its application in complex environment, its data security has been more and more attention. However, previous schemes have not adequately protected against Economic Denial of Sustainability (EDoS) attacks, where attackers can consume cloud resources and cost the payer a great deal of money. This lack of transparency and accountability is a major concern for data owners. In order to solve these issues, Fine grained control of access to Cloud Storage using CP-ABE Encryption Technique. The main aim is to secure encrypted cloud storage from EDoS attacks and provide resource consumption accountability by using CP-ABE schemes in a black-box manner that complies with the arbitrary access policies of CP-ABE. Many data owners choose to outsource encrypted data using Ciphertext-Policy Attribute-based Encryption (CP-ABE) for fine-grained access control. Two protocols are described for different settings and conducted performance. Security is analyzed to demonstrate the effectiveness and efficiency of the presented solution.

KEYWORDS: Cloud Computing, data storage, Security, Ciphertext-Policy Attribute-based Encryption, Access Control

.

I. INTRODUCTION

Cloud computing is an emergent technology in data analytics, which is used to retrieve, store and share big data in a distributed environment. Each day individuals and enterprises are storing their data in the server of the Cloud. The authorities of enterprises and individuals are starting to worry about the safety of big data in the Cloud.

The Cloud provides three types of services such as software, infrastructure and platform, but delivering the security to big data in the Cloud is the most difficult issue [1]. Generally, the government data, medical data and military data include sensitive details that need to be stored in the environment of the Cloud, but users are not sure about the security given by the service providers.

Cloud computing is a unique network or environment where access, maintenance and process can be done from any part of the world. It is a customized internet-based computer server. It is a current trend of modern technology. For the massive computational power, it is the best option for storing data. There is no uncertainty

that Cloud Data Server offers quick and solid types of assistance to its customers. When data is storing in the cloud storage the most important thing comes is the security of data. So, in recent years, cloud security is so much important issue because of the increasing of data [4].

Every day, the number of people using cloud computing services increases, and lots of data have been stored in cloud computing environments. Cloud computing has giant blessings that consist of remote storage, mobility, information sharing, value financial savings in hardware and software, etc [3]. The Cloud includes many advantages but still it lacks the security to store data. It is less popular to store the data in a single Cloud because of the failure of resource availability and also it includes some conditions where the inside malicious attackers will steal the data from a single Cloud. Data leakage to cloud services is also increasing every year because of attackers who are always trying to exploit the security vulnerabilities of cloud. Engineers and researchers try to identify the possible cloud threats and attacks in order to implement better security mechanisms to protect sensitive data and cloud computing environments [2].

Cryptography is the way to take care of the security worries of both users and service providers. Cryptography is the technique of encoding users' data to make it incomprehensible and impenetrable during storage or transmission. The very basic security threat that users face, while signing up for a cloud service, is giving open access to a service provider to their personal data. The second threat comes from other users in a shared virtual environment and the third security hazard is privileged access abuse from an outside source. Most cloud computing security solutions are related to cryptography of user's data on service provider's end so that no shared user or outside source can violate a particular user's personal data access rights.

Cryptography is the art of encoding secret information in illegible hidden format using an encryption key. The data is retrieved in its actual form on receiver end by decryption using the same secret key. Only the person with the secret key knowledge has access to the encrypted data and the right to decrypt it. The main ingredients of any cryptography process are: plain data, secret key, encryption algorithm, cipher data and decryption algorithm. Cryptography has two main types: symmetric or private or single key type cryptography and asymmetric or public key type cryptography [5].

Symmetric key cryptography algorithms are AES (Advanced Encryption Standard DES (Data Encryption Standard) 3DES (Triple Data Encryption Standard) IDEA and blowfish. The main issue is deliver the key to receiver into multi user application. These algorithm require low delay for data encode decode but provides low security. Public key cryptography algorithm is RSA and ECC (Elliptic Curve Cryptography) algorithm. Public and private keys are manipulated into public key cryptography algorithms. These algorithms accomplished high level security but increase delay for data encode and decode.

To solve these issues, Fine grained control of access to cloud storage using CP-ABE encryption technique. This technique involves encrypting the data and providing access to authorized users who can decrypt it using the access provided by the data owner. This approach will enable the fine granular access control and ensure security of the shared information. The rest of the work is organized as follows: The section

II describes the literature survey. The section III demonstrates the Fine grained control of access to cloud storage using CP-ABE encryption technique. The section IV evaluates the result analysis of presented technique. Finally the section V ends with conclusion.

II. LITERATURE SURVEY

R.Nivedhaa and J.Jean Justus et. al., [6] describes A Secure Erasure Cloud Storage system using Advanced Encryption Standard algorithm and Proxy Re-encryption. Proxy re-encryption scheme is suggested and combined with a distributed erasure code such that a secure and strong data storage and retrieval, but also lets a user to share his information on the cloud with a different user in the encrypted format itself. This work facilitates the use of encoding the encrypted files and sharing files in the encrypted format itself. This work uses the techniques of both encrypting and sharing the data. Erasure encoding supports sharing encrypted files and is valid in decentralized distributed system. A distributed erasure code is used to authorize the data safety in the dispersed cloud storage.

Rongzhi Wang et. al., [7] presents research on data security technology based on cloud storage. It introduces the implementation of DSBT (data secure storage scheme based on Tornado codes) system based on trusted log and research on data retrieval system as the core of the cloud storage prototype system. The system uses a simple three party security model, with Cassandra as the underlying distributed storage platform; the subsystem needs to carry out the logic module. The key part of this paper also gives a detailed flow chart and interface diagram. System performance test is also put in this part, first introduced the test parameters and the environment, and then for each function of the program design test case, the final result analysis.

Lalitha V.P, Sagar M.Y, Sharanappa S, Shredar Hanji, Swarup R et. al., [8] describes Data Security in Cloud. In this current work the data is stored in the server in encrypted fashion and only the admin is given the writes to decrypt the data. If an unauthorized user is trying to access any file or data from the cloud the admin can block the users IP address from accessing the data so that the security for data is given.

Keke Gai, Meikang Qiu, Hui Zhao et. al., [9] describes Security-Aware Efficient Mass Distributed Storage Approach for Cloud Systems in Big Data. a novel approach is presented that can efficiently split the file and separately store the data in the distributed cloud servers, in which the data cannot be directly reached by cloud service operators. The proposed scheme is entitled as Security-Aware Efficient Distributed Storage (SAEDS) model, which is mainly supported by the presented algorithms, named Secure Efficient Data Distributions (SED2) Algorithm and Efficient Data Conflation (EDCon) Algorithm. the experimental evaluations have assessed both security and efficiency performances.

Yibin Li, Keke Gai, Longfei Qiu, Meikang Qiu, Hui Zhao et. al., [10] describes Intelligent cryptography approach for secure distributed big data storage in cloud computing. This scheme is entitled Security-Aware Efficient Dis- tributed Storage (SA-EDS) model, which is mainly supported by our proposed algorithms, including Alternative Data Distribution (AD2) Algorithm , Secure Efficient Data Distributions (SED2) Algorithm and Efficient Data Conflation (EDCon) Algorithm. The experimental evaluations have assessed both security and efficiency

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

performances and the experimental results depict that our approach can effectively defend main threats from clouds and requires with an acceptable computation time.

III. FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE

In this section, Fine grained control of access to cloud storage using CP-ABE encryption technique is presented. The Fig. 1 shows the architecture of presented Fine grained access control to cloud storage using CP-ABE encryption technique. The main objective of this project is to enhance the security of encrypted cloud storage by preventing attacks and ensuring accountability for resource consumption. The system will enable access to data only by authorized users, rather than making it available to everyone.

Data owners: In this module, Data owners are the owner and publisher of files and pay for the resource consumption on file sharing. As the payers for cloud services, the data owners want the transparency of resource consumption to ensure fair billing. The data owners require the cloud provider to justify the resource usage. In our system, the data owner is not always online.

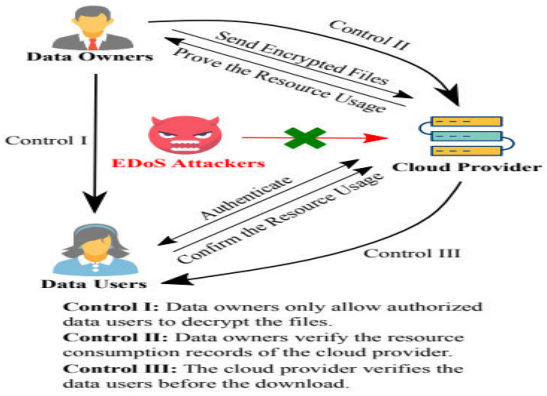


Fig. 1: The Architecture of Fine grained control of access to cloud storage using CP-ABE encryption technique

Data users: In this module users want to obtain some files from the cloud provider stored on the cloud storage. They need to be authenticated by the cloud provider before the download (to thwart EDoS attacks). The authorized users then confirm (and sign for) the resource consumption for this download to the cloud provider.

Cloud provider: Cloud provider hosts the encrypted storage and is always online. It records the resource consumption and charges data owners based on that record. The cloud is not public-accessible in our system as it has an authentication based access control. Only data users satisfying the access policy can download the corresponding files. The cloud provider also collects the proof of the resource consumption to justify the billing.

Security against EDoS Attacks: EDoS attackers are those that do not satisfy the access policy (i.e., unauthorized users) but want to trigger the cloud provider to send something through the network, as a result the resource consumption increases. To thwart such attacks, the cloud provider uses authentication. The protocols only send a constant amount of bytes to the data user before it passes the cloud-side access control. To succeed a EDoS attack in our definition, the attacker has to first pass the cloud-side access control.

The main focus of this work is to implement Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to enable fine-grained and owner-centric access control for sharing encrypted files with other users. The project utilizes Partially Outsourced Protocols (POP) and Fully Outsourced Protocols (FOP) to ensure accountability for resource consumption.

Attribute-based encryption (ABE) is a type of encryption that allows for access control based on attributes like user characteristics or data properties. This is different from traditional access control methods, such as

user names or roles. ABE can provide greater confidentiality compared to other methods, due to its ability to offer fine-grained control over who can access encrypted data. ABE is highly flexible and efficient, making it applicable to various domains and applications. There are several types of ABE schemes, including key-policy ABE. ABE is making crucial advances in solving problems related to confidentiality, and its versatility and efficiency make it an attractive solution for many use cases.

The security requirements of the system are achieved through two key components: 1. A cloud-side access control mechanism that blocks users whose attribute set does not meet the access policy A; 2. A proof-collecting subsystem that allows the cloud provider to collect proofs of resource consumption from users and present them to the data owner later.

Real-world scenarios often involve specifying a maximum expected download time, and data owners can remain offline unless they choose to increase this value. The first protocol, Partially Outsourced Protocol (POP), is designed to address such scenarios. In cases where the data owner cannot set expectations for download times or would be offline for an extended period, the Fully Outsourced Protocol (FOP) can be utilized, allowing the data owner to delegate control to the cloud.

The Partially Outsourced Protocol (POP) involves encrypting an ephemeral key in CP-ABE by the data owner. This key is then used for both message encryption/decryption and cloud-side access control. The data owner provides the cloud provider with a set of N challenge cipher texts $\{enchali\}\in[N]$ and the corresponding hashed challenges $\{hashi\}i\in[N]$.

To prove legitimacy to the cloud provider, the user must show that the decryption result $chalj$ of a randomly selected unused challenge ciphertext $enchalj$ is a pre-image of $hash$. If the user's response is valid, the cloud provider stores the response for further resource consumption accounting. To reduce storage space and improve efficiency, a bloom filter can be introduced for data owners to store their challenge plaintexts. This bloom filter can be stored locally or remotely on the cloud server. As the challenge update process cannot be outsourced to the cloud and must be implemented on demand or periodically by the data owner, the scheme is referred to as the Partially Outsourced Protocol (POP).

The procedure of POP is done in 4 phases which are as follows: i) Encrypt and Upload (POP-EU): This operation is implemented by an individual data owner independently. ii) Cloud-side Access Control: POP-CR. POP-CR-1: The cloud provider selects one of the unused challenge and sends the following tuple to the user. iii) Challenge update (POP-SU): The scheme allows for on-demand or periodic challenge updates by the data owner, as long as the specified upper bound of download times (N) has not yet been reached. If the data owner wishes to provide additional challenges, they can do so by being online for a short period. The update process is similar to that in the POP-EU-2 phase, using the same key k. The data owner is assumed to keep a record of session keys either in local storage or encrypted form outsourced to the cloud. As the plaintext space for challenges is sufficiently large, it is assumed that no duplicated challenge plaintexts are generated. If a bloom filter is used (and its encrypted form), it will need to be reconstructed in this case. iv) Resource Accounting (POP-RA): Data

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

owners and the cloud interactively implement this operation

The Fully Outsourced Protocol (FOP) is a protocol that allows for outsourced challenge generation and update, as well as resource accounting, without relying on an external PKI. It is based on the signature algorithm and offers two key differences when compared to the Proof of Possession (POP) protocol. Firstly, in FOP, the cloud provider generates the challenges { ⟦enchal⟧ i }(i∈[N]) instead of the data owners. Secondly, the data owners generate a pair of signature keys (vk, sk) for each file, which users can use to sign a confirmation to prove resource consumption.

The FOP procedure involves four steps: i) Encrypt and Upload (FOP-EU): The data owner encrypts the file using a symmetric key and uploads it to the cloud. The data owner also generates a pair of signature keys (vk, sk) for the file. ii) Outsourced Challenge Generation (FOP-CG): The cloud provider generates the challenges { ⟦enchal⟧ i }(i∈[N]) for the file, which are then sent to the data owner. This step can be done in advance or on demand. iii)Challenge-Response (FOP-CR): In this step, the data owners and the cloud run the operation. The data owner calculates a response to the challenges sent by the cloud and sends them back. iv) Resource Accounting (FOP-RA): Legitimate users can sign a confirmation using the signature keys (vk,sk) to prove resource consumption. This operation is interactively implemented. Overall, FOP provides a secure and efficient solution for outsourced challenges generation/update and resource accounting, without relying on an external PKI (Public Key Infrastructure.

IV. RESULT ANALYSIS

In this section, Fine grained control of access to cloud storage using CP-ABE encryption technique is implemented. The Fig. 2 shows the login page of presented approach.

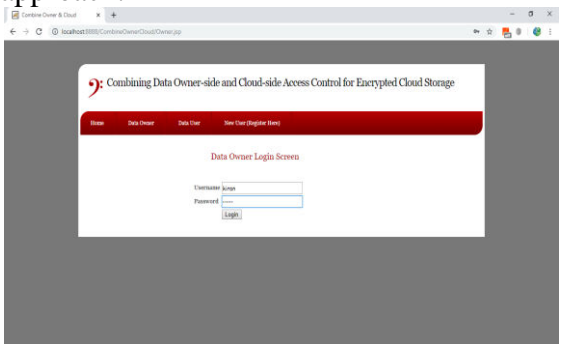


Fig. 2: Log-in Page

After the login the data owner will get access to upload the files in cloud storage. The Fig. 3 shows the files uploading page.

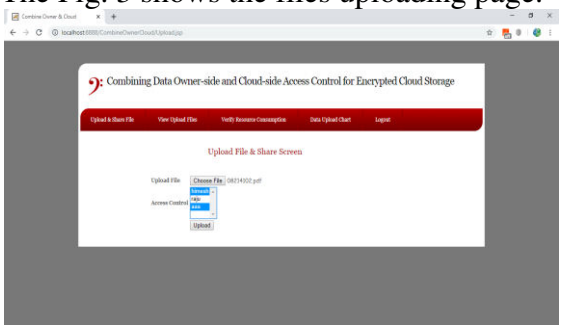


Fig. 3: Uploading files to cloud storage

The fig. 4 shows the data file screen.

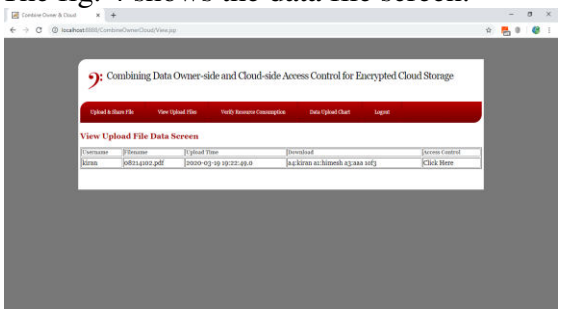


Fig. 4: Data File Screen

All uploaded files store in encrypted format inside "WEB-INF/user" folder. In above screen we can see ACCESS POLICY also generated by data owner. Now click on 'View Resource Consumption' link to request cloud to give proof on resource consumption which means how many users access this data owner's file. By using this option we can

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

prevent cloud from cheating or applying fraud resource consumption cost. The Fig. 5 shows the file requesting page.

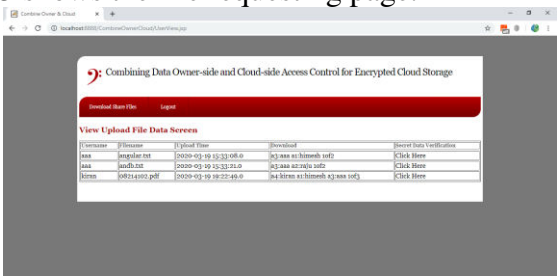


Fig. 5: File requesting page

In above screen data user can see all files from all data owners shared with him. By clicking on 'Click Here' link user can request cloud for file download and will get below screen.

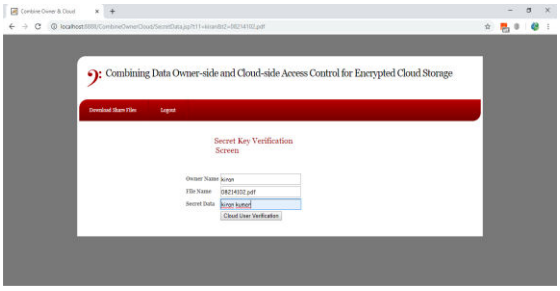


Fig. 6: Secret Challenge Page

In above screen cloud is asking data user for secret data challenge and if user give correct data owner secret data then only file will be downloaded otherwise not. If the user enters wrong data then the verification is failed which is shown in Fig. 7.

Fig. 7: Verification Failed

From Fig. 7 it is clear that secret data verification failed at cloud side and cloud will not allow user to download file if wrong data is entered.

V. CONCLUSION

In this work, Fine grained control of access to cloud storage using CP-ABE encryption technique is presented. The main aim is to enhance the security of encrypted cloud storage by combining access control measures both at the cloud and data owner sides. This approach is designed to resist Distributed Denial of Service (DDoS) and Extreme Denial of Service (EDoS) attacks while also providing resource consumption accounting. This system supports arbitrary Conjunctional Policy Attribute-Based Encryption (CP-ABE) constructions, which are secure against malicious data users and covert cloud providers. We have relaxed the security requirements for the cloud provider to accommodate covert adversaries, a more practical and relaxed notion compared to semi-honest adversaries. To optimize resource consumption accounting and reduce overhead, we utilize bloom filters and probabilistic checks. From the result analysis, it is observed that this approach offers very less overhead compared to existing systems.

VI. ACKNOWLEDGEMENT

We thank CMR Technical Campus for supporting this paper titled "Fine Grained Control of access to cloud storage using CP-abe Encryption Technique", which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans, Head Of the Department, Department Of Computer Science and Engineering, Guide and Teaching and Non- Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work

VII. REFERENCES

[1] Mohan Naik Ramachandra, Madala Srinivasa Rao, Wen Cheng Lai, Bidare Divakarachari Parameshachari,

Jayachandra Ananda Babu and Kivudujogappa Lingappa Hemalatha, "An Efficient and Secure Big Data Storage in Cloud Environment by Using Triple Data Encryption Standard", Big Data Cogn. Comput. 2022, 6, 101, doi:10.3390/bdcc6040101

[2] Amr M. Sauber Passent M. El-Kafrawy, Amr F. Shawish , Mohamed A. Amin, and Ismail M. Hagag, "A New Secure Model for Data Protection over Cloud Computing", Hindawi Computational Intelligence and Neuroscience Volume 2021, Article ID 8113253, 11 pages, doi:10.1155/2021/8113253

[3] Mrs. Anjali Sharma, Dr. Garima Sinha, "An Efficient Approach on Data Security with Cloud Computing Environment: A Comprehensive Research", Turkish Journal of Computer and Mathematics Education Vol.12 No.14 (2021), 1372 – 1382

[4] Md. Alamgir Hossain & Md. Abdullah Al Hasan, "Improving cloud data security through hybrid verification technique based on biometrics and encryption system", International Journal of Computers and Applications, 2020, DOI: 10.1080/1206212X.2020.1809177

[5] Sameer A. Nooh, "Cloud Cryptography: User End Encryption", 2020 International Conference on Computing and Information Technology, University of Tabuk, Kingdom of Saudi Arabia. Volume: 01, Issue: ICCIT- 1441, Page No.: 397 - 400, 9th & 10th Sep. 2020

[6] R.Nivedhaa and J.Jean Justus, "A Secure Erasure Cloud Storage system using Advanced Encryption Standard algorithm and Proxy Re-encryption", International Conference on Communication and Signal Processing, April 3-5, 2018, India

[7] Rongzhi Wang, Research on data security technology based on cloud storage", 13th Global Congress on Manufacturing and Management, GCMM 2017, Elsevier, doi: 10.1016/j.proeng.2017.01.286

[8] Lalitha V.P, Sagar M.Y, Sharanappa S, Shredar Hanji, Swarup R, "Data Security in Cloud", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017)

[9] Keke Gai, Meikang Qiu, Hui Zhao, "Security-Aware Efficient Mass Distributed Storage Approach for Cloud Systems in Big Data", 2016 IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, 978-1-5090-2403-2/16, DOI 10.1109/Big Data Security

[10] Yibin Li, Keke Gai, Longfei Qiu, Meikang Qiu, Hui Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing", Information Sciences 0 0 0 (2016) 1–13, doi:10.1016/j.ins.2016.09.005

Sai Preethi is currently pursuing B. Tech final year in the stream of Computer Science and Engineering in CMR Technical Campus, Medchal, Hyderabad, Telangana, India.

Preethi is currently pursuing B. Tech final year in the stream of Computer Science and Engineering in CMR Technical Campus, Medchal, Hyderabad, Telangana, India.

Chandrika is currently pursuing B. Tech final year in the stream of Computer Science and Engineering in CMR Technical Campus, Medchal, Hyderabad, Telangana, India.

Ms. Rakshitha Okali is working as an Assistant professor in the Department of Computer science and Engineering, CMR Technical Campus, Medchal, Hyderabad. She is having one year of Teaching Experience.

10. CERTIFICATES

IMPACT FACTOR
7.812

International Journal for Innovative Engineering and Management Research

(ISSN 2456 - 5083)

CERTIFICATE

ELSEVIER SSRN

DOI: 10.48047/IJIEMR/V12/I03/52

This is to Certify that Prof./Dr./Mr./PINNINTI SAI PREETHI From CMR Technical Campus, Medchal, Hyderabad, Telangana, India.Participated in the International Journal of Engineering and Management Research.Presented a Paper Entitled"FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE USING CP-ABE ENCRYPTION TECHNIQUE"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-03 Mar-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com

IMPACT FACTOR
7.812

International Journal for Innovative Engineering and Management Research

(ISSN 2456 - 5083)

ELSEVIER
SSRN

CERTIFICATE

DOI: 10.48047/IJIEMR/V12/I03/52

This is to Certify that Prof./Dr./Mr./NANAM PREETHI From CMR Technical Campus, Medchal, Hyderabad, Telangana, India.Participated in the International Journal of Engineering and Management Research.Presented a Paper Entitled"FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE USING CP-ABE ENCRYPTION TECHNIQUE"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-03 Mar-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com

IMPACT FACTOR
7.812

International Journal for Innovative Engineering and Management Research

(ISSN 2456 - 5083)

CERTIFICATE

ELSEVIER
SSRN

DOI: 10.48047/IJIEMR/V12/I03/52

This is to Certify that Prof./Dr./Mr./AMARTHALURI CHANDRIKA From CMR Technical Campus, Medchal, Hyderabad, Telangana, India.Participated in the International Journal of Engineering and Management Research.Presented a Paper Entitled"FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE USING CP-ABE ENCRYPTION TECHNIQUE"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-03 Mar-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561

Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081

info.ijiemr@gmail.com

IMPACT FACTOR 7.812

International Journal for Innovative Engineering and Management Research

(ISSN 2456 - 5083)

CERTIFICATE

DOI: 10.48047/IJIEMR/V12/I03/52

This is to Certify that Prof./Dr./Mr./RAKSHITHA OKALI From CMR Technical Campus, Medchal, Hyderabad, Telangana, India.Participated in the International Journal of Engineering and Management Research.Presented a Paper Entitled"FINE GRAINED CONTROL OF ACCESS TO CLOUD STORAGE USING CP-ABE ENCRYPTION TECHNIQUE"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-03 Mar-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com