A

Major Project Report on

**DUAL ACCESS CONTROL FOR CLOUD BASED DATA STORAGE**

**AND SHARING**

submitted in partial fulfillment of the requirements for the award of the

B.Tech Degree

By

B.SIDDARTHA

(20EG105606)

P.VIVAAN PRAKASH

(20EG105638)

D.RITHWIK REDDY

(20EG105645)



Under the guidance of

K.Sadanandam

Assistant professor,CSE

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ANURAG UNIVERSITY**

**VENKATAPUR-500088**

**TELANGANA**

**YEAR 2023-2024**

**DECLARATION**

We here by declare that the Report entitled “**DUAL ACCESS CONTROL FOR CLOUD BASED DATA STORAGE AND SHARING**” submitted for the award of Bachelor of technology Degree is our original work and the report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

Date:

B.SIDDARTHA

(20EG105606)

P.VIVAAN PRAKASH

(20EG105638)

D.RITHWIK REDDY

(20EG105645)

**CERTIFICATE**

This is to certify that the Report entitled “**DUAL ACCESS CONTROL FOR CLOUD BASED DATA STORAGE AND SHARING**”,that is being submitted by **B.Siddartha** bearing roll number **20EG105606** **P.Vivaan Prakash** bearing roll number **20EG105638**,**D.Rithwik Reddy** bearing roll number **20EG105645** in partial fulfillment for the award of B.Tech in Computer Science And Engineering to the Anurag University is a record of bonafide work carried out by them under our guidance and supervision.

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

## 

Dean,CSE

Signature of supervisor

K.Sadanandam

Assistant Professor ,CSE

External Examiner

**ACKNOWLEDGEMENT**

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mr K.Sadanandam** for his constant encouragement and inspiring guidance without which this project could not have been completed.His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like express our special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University,for their encouragement and timely support in our B.Tech program.

We would like acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy** , Dean, Dept. of CSE ,Anurag University, We also express our deep sense of gratitude to **Dr. V V S S S Balaram** ,Acadamic coordinator, **Dr .Pallam Ravi** , Project in-Charge ,**Dr.A.Jyothi** ,Project Co-ordinator and Project review committee members,whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of our project work.

**ABSTRACT**

Cloud-based data storage service has drawn increasing interests from both academic and industry in the recent years due to its efficient and low cost management. Since it provides services in an open network, it is urgent for service providers to make use of secure data storage and sharing mechanism to ensure data confidentiality and service user privacy. To protect sensitive data from being compromised, the most widely used method is encryption. However, simply encrypting data (e.g., via AES) cannot fully address the practical need of data management. Besides, an effective access control over download request also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service. In this paper, we consider the dual access control, in the context of cloud-based storage, in the sense that we design a control mechanism over both data access and download request without loss of security and efficiency. Two dual access control systems are designed in this paper, where each of them is for a distinct designed setting. The security and experimental analysis for the systems are also presented.

|  |  |  |
| --- | --- | --- |
| S.No | CONTENT | **Page No.** |
| 1 | 1. Introduction   1.1Problem statement  1.2Problem Ilustration  1.3 Objective | 1-2  2  2-3 |
| 2 | 2 . Literature Survey | 4 |
| 3 | 3. Proposed method  3.1Existing method disadvantages  3.2Advantagesof proposed system  3.3System Requirements  3.3.1 Hardware Requirements  3.3.2 Software  Requirements | 5  5-6  6-7  7  7  7 |
| 4 | 4. Design  4.1 Use Case Diagram  4.2 Sequence Diagram  4.3 Activity Diagram   * 1. Class Diagram | 8  9  10  11 |
| 5 | 5. Implementation  5.1 Introduction  5.2 Advanced Encryption Standarad  5.3 Introduction to technologies used   * 1. Sample code | 12  12  12-13  13-17  18-20 |
| 6 | 6. Experiment Results | 21-27 |
| 7 | 7. Discussion Of Results  7.1 Experiment1  7.2 Experiment2  7.3 Experiment3  7.4 Test cases | 28  28-29  29  30 |
| 8 | 8. Conclusion | 31 |
| 9 | 9. Future Scope | 32 |
| 10 | 10. References | 33-34 |

**INDEX**

**LIST OF SCREENS**

|  |  |  |
| --- | --- | --- |
| Figure No. | Figure Name | Page No |
| 6.1 | Home Screen | 21 |
| 6.2 | Login Page | 21 |
| 6.3 | Owner registration page | 22 |
| 6.4 | Owner file upload page | 22 |
| 6.5 | Owner uploaded files | 23 |
| 6.6 | User Registratin Page | 23 |
| 6.7 | User Searching files | 24 |
| 6.8 | User requesting file key | 24 |
| 6.9 | Authority Login | 25 |
| 6.10 | User Requested files | 25 |
| 6.11 | Access control policy page | 26 |
| 6.12 | User entering secret key | 26 |
| 6.13 | Requested file details | 27 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Fig no. | Figure Name | Page no |
| 4.1.1 | Use Case Diagram | 8 |
| 4.2.1 | Sequence Diagram | 9 |
| 4.3.1 | Activity Diagram | 10 |
| 4.4.1 | Class Diagram | 11 |

**LIST OF GRAPHS**

|  |  |  |
| --- | --- | --- |
| s.no | Graphs | Page no |
| 1 | Graph 7.1.1 Comparision in terms of compuatational cost | 28 |
| 2 | Graph 7.2.1 Comparision in terms of communication cost | 28 |
| 3 | Graph 7.3.1 Comparision of CP-ABE in terms of computational cost. | 29 |

**1.INTRODUCTION**

In the recent decades, cloud-based storage service has attracted considerable attention from both academia and industries. It may be widely used in many Internet-based commercial applications (e.g., Apple iCould) due to its long-list benefits including access flexibility and free of local data management. Increasing number of individuals and companies nowadays prefer to outsource their data to remote cloud in such a way that they may reduce the cost of upgrading their local data management facilities/devices. However, the worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service.

Since the cloud (which is deployed in an open network) is not be fully trusted, it is generally recommended to encrypt the data prior to being uploaded to the cloud to ensure data security and privacy. One of the corresponding solutions is to directly employ an encryption technique (e.g., AES) on the outsourced data before uploading to cloud, so that only specified cloud user (with valid decryption key) can gain access to the data via valid decryption

In a cloud-based storage service, there exists a common attack that is well-known as resource-exhaustion attack. Since a (public) cloud may not have any control over download request (namely, a service user may send unlimited numbers of download request to cloud server), a malicious service user may launch the denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks to consume the resource of cloud storage service server so that the cloud service could not be able to respond honest users’ service requests. As a result, in the “pay-as-you-go” model, economic aspects could be disrupted due to higher resource usage. The costs of cloud service users will rise dramatically as the attacks

scale up. This has been known as Economic Denial of Sustainability (EDoS) attackswhich targets to the cloud adopter’s economic resources. Apart from economic loss, unlimited download itself could open a window for network attackers to observe the encrypted download data that may lead to some potential information leakage (e.g., file size). Therefore, an effective control over download request for outsourced (encrypted) data is also needed.

In this paper, we propose a new mechanism, dubbed dual access control, to tackle the above aforementioned two problems. To secure data in cloud-based storage service, attribute-based encryption (ABE) is one of the promising candidates that enables the confidentiality of outsourced data as well as fine-grained control over the outsourced data. In particular, Ciphertext-Policy ABE (CP-ABE) provides an effective way of data encryption such that access policies, defining the access privilege of potential data receivers, can be specified over encrypted data. Note that we consider the use of CP-ABE in our mechanism in this paper. Nevertheless, simply employing CP-ABE technique is not sufficient to design an elegant mechanism guaranteeing the control of both data access and download request.

* 1. **PROBLEM STATEMENT**

In recent years, cloud-based data storage and sharing has become increasingly popular, as it offers numerous benefits such as scalability, accessibility, and cost-effectiveness. However, the security and privacy of cloud-based data storage and sharing remain a significant concern, as sensitive data is often stored in the cloud. One of the critical challenges is ensuring that only authorized users have access to the data while still enabling data sharing among authorized users. To address this challenge, dual access control mechanisms have been proposed, which provide separate controls for data owners and data users. In this approach, the data owner has control over who can access and modify the data, while the data user has control over who they can share the data with. This approach offers a flexible and scalable solution for secure and controlled data sharing in the cloud. This paper aims to explore the concept of dual access control for cloud-based data storage and sharing, its benefits and limitations, and existing research in this area**.**

* 1. **PROBLEM ILLUSTRATION**

Sample Values:Authority: Hospital IT department registers healthcare professionals with unique identifiers (e.g., doctor123, nurse456) and assigns specific access privileges based on roles (e.g., doctors have access to all patient records, nurses have access to specific patient records).

Data Owner (Hospital): The hospital uploads patient medical records to the cloud, encrypted using cipher text policy with attribute-based encryption (ABE).(e.g., role = doctor).

Email:user@gmail.com, password:xxxxxx, Access policy :Read or Download , upload file :abxyz

Data User (Healthcare Professionals): Authorized healthcare professionals access patient medical records stored in the cloud by sending download requests. (eg. role =patient).

Email:user@gmail.com , password: xxxxx, Access policy :Read , Decryption

key:xxxxxxxx ,File:abxyz.

Cloud: Stores encrypted patient medical records securely and handles download requests from authorized healthcare professionals.

* 1. **OBECTIVE**

**Enhance Data Security:** Go beyond basic encryption to provide stronger protection for sensitive information stored in the cloud.

**Granular Access Control:** Implement a two-level control mechanism that allows for both cloud service provider oversight and user-defined access restrictions.

**Prevent Denial-of-Service Attacks:** Establish control over download requests to prevent malicious attempts at disrupting user access to data.

**Maintain Efficiency:** Achieve the desired security improvements without significantly impacting the performance of data access and management.

**2.LITERATURE SURVEY**

Xue et al. [38] stated that the previous works could not fully defend the EDoS attack in the algorithmic (or protocol) level, and they further proposed a solution to secure cloud data sharing from the attack. However, [38] suffers from two disadvantages. First, the data owner is required to generate a set of challenge ciphertexts in order to resist the attack, which enhances its computational burden. Second, a data user is required to decrypt one of the challenge ciphertexts as a test, which costs a plenty of expensive operations (e.g., pairing). Here the computational complexity of both parties is inevitably increased and meanwhile, high network bandwidth is required for the delivery of ciphertexts. The considerable computational power of cloud is not fully considered . In this paper, we will present a new solution that requires less computation and communication cost to stand still in front of the EDoS attack.

Antonis Michalas [20] proposed a data sharing protocol that combines symmetric searchable encryption and ABE, which allows users to directly search over encrypted data. To implement the functionality of key revocation in ABE, the protocol utilizes SGX to host a revocation authority ,ABE has two main research branches: one is CP-ABE, and the other is KP-ABE which refers to as keypolicy ABE. This paper mainly deals with the former. In a CP-ABE, decryption key is associated with attribute set and ciphertext is embedded with access policy. This feature makes CP-ABE quite suitable for secure cloud data sharing this is so because KP-ABE requires decryption key to be associated with access policy which yields heavy storage cost for cloud user. Since the introduction of seminal CP-ABE [9], many works have been proposed to employ CP-ABE in various applications.

Bakas and Michalas [3] later extended the protocol in [20] and proposed a hybrid encryption scheme that reduces the problem of multi-user data sharing to that of a single-user. In particular, the symmetric key used for data encryption is stored in an SGX enclave, which is encrypted with an ABE scheme. Similar to [20], it deals with the revocation problem in the context of ABE by employing the SGX enclave. In this work, we employ SGX to enable the control of the download request (such that the DDoS/EDoS attacks can be prevented). In this sense, the purpose and the technique of ours are different from that of the protocols.

**3.PROPOSED SYSTEM**

In this paper, we propose a new mechanism, dubbed dual access control, to tackle the above aforementioned two problems. To secure data in cloud-based storage service, attribute-based encryption (ABE) is one of the promising candidates that enables the confidentiality of outsourced data as well as fine-grained control over the outsourced data. In particular, Ciphertext-Policy ABE (CP-ABE) provides an effective way of data encryption such that access policies, defining the access privilege of potential data receivers, can be specified over encrypted data. Note that we consider the use of CP-ABE in our mechanism in this paper. Nevertheless, simply employing CP-ABE technique is not sufficient to design an elegant mechanism guaranteeing the control of both data access and download request.

The systems we propose are with the following distinct features:

(1) Confidentiality of outsourced data: In our proposed systems, the outsourced data is encrypted prior to being uploaded to cloud. No one can access them without valid access rights.

(2) Anonymity of data sharing :Given an outsourced data, cloud server cannot identify data owner, so that the anonymity of owner can be guaranteed in data storage and sharing.

(3)Efficiency: Despite the additional layer of access control, the system maintains efficiency in data management operations. It achieves this through optimized algorithms and streamlined processes, minimizing latency and resource overhead.

(4) Dual access control mechanism: Unlike traditional access control systems that focus solely on data access, this system extends control to download requests as well. By regulating both aspects, it effectively mitigates the risk of EDoS attacks and enhances overall security posture.

**3.1 Existing methods Disadvanatges:**Single-Layer Access Control: Many existing methods may only focus on one aspect of access control, such as data encryption, without considering the need for controlling download requests or preventing EDoS attacks.Limited Security Measures: Some existing methods may rely solely on basic encryption techniques, such as AES, which may not provide sufficient protection against sophisticated attacks or insider threats.Lack of Flexibility: Some existing methods may not offer flexibility in adapting to different use cases or settings, limiting their applicability in diverse environments. Vulnerability to EDoS Attacks: Existing methods may lack adequate defenses against EDoS attacks, which can disrupt service availability by overwhelming the system with download requests.Data Confidentiality and Privacy:Consider a healthcare organization that stores patient medical records in the cloud. Without proper encryption measures, sensitive patient information, such as diagnoses and treatment histories, could be vulnerable to unauthorized access.This poses significant privacy concerns and may lead to breaches of patient confidentiality.Access Control Over Download Requests: Imagine an online file-sharing platform where users can download files stored in the cloud. If there are no controls in place to regulate download requests, malicious actors could launch EDoS attacks by flooding the system with a large number of simultaneous download requests.**3.2.ADVANTAGES PROPOSED SYSTEM**

**Enhanced Security:** By implementing both data access and download request controls, the system provides a comprehensive security framework. This significantly reduces the risk of unauthorized access, data breaches, and EDoS attacks, thereby safeguarding sensitive information.

**Improved Privacy Protection:**Users' privacy is prioritized through stringent access control mechanisms, ensuring that only authorized individuals can access and download data. This fosters trust among users and mitigates concerns regarding data privacy and confidentiality.

**Mitigation og EDOS Attacks:** The inclusion of access controls for download requests helps prevent EDoS attacks aimed at disrupting service availability. By regulating the rate and frequency of downloads, the system ensures equitable resource allocation and maintains service continuity.

**Efficient Data Management:**Despite the added security measures, the system maintains efficiency in data management operations. Optimized algorithms and streamlined processes minimize latency and resource overhead, enabling seamless access and retrieval of data.

**Scalability:** The system is designed to scale with growing data volumes and user demands, ensuring continued performance and security as the cloud environment expands. This scalability accommodates the dynamic nature of cloud-based services and supports organizational growth.

**3.3 SYSTEM REQUIREMENTS**

**3.3.1 HARDWARE REQUIREMENTS:**

Processor : i3

Ram : 4GB

Hard Disk : 500GB

**3.3.2 SOFTWARE REQUIREMENTS:**

Programming Language : Java

Technologies : Sevlets,jsp,jdbc

Front End Tech : Html,css,js

Web Server : Tomcat Sever

Database Server : MySql Database Server

Database Client : SQLyog

IDE : Eclipse

**4. DESIGN**

**4.1 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.



Fig 4.1.1 Use case Diagram

**4.2 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



Fig 4.2.1 Sequence Diagram

**4.3 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Fig 4.3.1 Activity Diagram

**4.4 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



Fig 4.4.1 Class Diagram

**5.IMPLEMENTATION**

**5.1 Introduction**

We employ the use of a hybrid system to protect the data, which combines the efﬁciency of a symmetric-key system with the convenience of a public-key system. In particu- lar, the proposed dual access control systems are both in Key/Data Encapsulation Mechanism (KEM/DEM) setting Themessageisencryptedbyanefﬁcientsymmetric-key encryption scheme, while the inefﬁcient public-key scheme (i.e., the CP-ABE) is used only to encrypt/decrypt a short key value. To achieve the security requirements of anonymous data sharing, conﬁdentiality of shared data and access control on shared data, we employ the CP-ABE technique as the basic building block. Speciﬁcally, we present the construction based on theCP-ABE scheme in due to its efﬁciency and elegant construction. To achieve the security requirements of anonymous download request and access control on download request, we design an effective mechanism that the cloud can judge whether a data user is authorized or not without revealing any sensitive information (including the identity of the data user, the plaintext of the outsourced data)toit.Intheﬁrstsystem,thecloudneedsthehelpofthe authority during the judgement on the download request (sent by a data user). As a result, the authority needs to be always online.

**5.2 Advanced Encryption standard:**

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix .

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.



Fig.5.2.1 AES Structure

**5.3 INTRODUCTION TO TECHNOLOGIES USED**

**Java**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages**.**

Front-end web development: The front-end technology interacts with the user and Java interface. It helps to insert and submit data. Java web development uses JavaServer Pages or JSP for the front-end form or table.

Backend web development: The backend technology maintains and updates data of the database. Java uses Servlet, spring, and other advanced technology.

Database management handles or fetches data from the database using the Java database driver. The Java technology uses JDBC, Hibernate to handle the database. Types of the Java Web Technologies.

**Servlet API**

(JAVA Web application programming interface) Servlet, filter, filter chain, servlet config, and other interfaces are available in the javax. Servlet package. The capabilities of servers that host apps are increased by using Servlet. The request-response model is used in web development applications written with Java servlets. From initialization to garbage collection, a servlet has a life cycle. Servlets are useful for various tasks, including collecting data via web page forms, presenting data from a database or any other third-party source, etc. Servlets are Java programs that run on a web application and send client requests to databases or servers. After talking with the database, the servlets help process the client's request and provide results.

**JSP**

Developers employ JavaServer Pages or JSP technology to quickly produce platformand server-independent online content. Normally, the developer works on separate Common Gateway Interface files to embed dynamic elements in HTML pages. Java JSP technology can be used, as it has access to the whole Java API family. The JSP technology pieces code to control web information and moves dynamically. A JSP page comprises static data written in HTML, WML, XML, and other markup languages. Special JSP tags simplify Java code into HTML pages, making web development user-friendly. The JSP technology allows embedding bits of servlet code in a text-based document. JSP is a popular Java EE technology that allows programmers to create complex dynamic web pages quickly.

**JDBC Driver**

or Java Database Connectivity JDBC Driver is a connector between database and Java web application. Java database connectivity helps to update and modify data using queries. The jdbc driver is an essential part of Java web development. This driver helps to send data to the database and retrieve data from the database. JDBC is a set of methods and queries for accessing databases written in Java. Clients can use web applications using JDBC drivers to update any information in the database. JDBC drivers connect to databases in four ways: JDBC-ODBC Bridge Driver, Network Protocol Driver, Native Driver, and Thin Driver. Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java. For web development, the Java Persistence API employs object-relational mapping. This mapping connects a database to an object-oriented model. Java Persistence makes it simple to manage relational data in Java web applications. The Java Persistence API aids in database data management. This API sends data to a database and retrieves data from it regularly.

Large amounts of code, proprietary frameworks, and other files are not required. JPA gives a straightforward technique of database communication. A database is an object relational approach for interacting with Java web development. JPA is a set of lightweight classes and methods for interacting with databases.

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

**HTML(HyperText Markup Language):** HTML is the standard markup language for creating web pages and web applications. It provides the structure and content of a webpage through various elements and tags. HTML, or Hypertext Markup Language, stands as the foundational language for structuring content on the web. It employs a system of tags enclosed in angle brackets to define elements, forming a hierarchical structure within documents. These elements encompass a wide range of content types, from headings and paragraphs to lists, links, and multimedia. Attributes, which provide additional information or behavior specifications, can be added to elements. Embracing semantic HTML ensures clarity, accessibility, and SEO optimization.

**CSS (Cascading Style sheets):** CSS is a style sheet language used to describe the presentation of a document written in HTML. It defines the layout, colors, fonts, and other visual aspects of a webpage, enhancing its appearance and user experience. It advocates for the separation of concerns, keeping the structure (HTML), presentation (CSS), and behavior (JavaScript) distinct for easier maintenance and scalability. With concepts like cascading, specificity, and the box model, CSS allows for flexible and responsive designs that adapt seamlessly across various devices and screen sizes.

**Tomcat Server**

Apache Tomcat, developed by the Apache Software Foundation, is an open-source web server and servlet container widely chosen for deploying Java-based web applications. Serving as both a servlet container and an HTTP server, it facilitates the execution of Java servlets and JavaServer Pages (JSP) while also delivering static content like HTML, CSS, and JavaScript files. Tomcat adheres to Java EE specifications, ensuring compatibility with a plethora of Java-based technologies and frameworks. Its modularity allows for flexible configuration and extension, supporting features like clustering for scalability and built-in security mechanisms such as SSL/TLS support and access control. With comprehensive documentation and a vibrant community, Tomcat offers robustness, scalability, and platform independence, making it a preferred choice among developers and organizations for hosting dynamic web applications.

**JavaScript:** JavaScript is a high-level programming language that enables interactive and dynamic features on web pages. It allows developers to manipulate the content and behavior of web pages in response to user actions, such as clicking buttons, entering text, or scrolling. JavaScript is a versatile and powerful programming language primarily used for creating dynamic and interactive web content. Developed in the mid-1990s by Brendan Eich at Netscape Communications Corporation, JavaScript has since become one of the most widely used programming languages in the world.

It is an integral part of web development, enabling developers to add functionality and interactivity to websites and web applications. JavaScript is a high-level, interpreted language, meaning that it is executed by the browser's JavaScript engine without the need for compilation. It is known for its lightweight syntax, making it accessible to both beginners and experienced developers alike. JavaScript supports both object-oriented and functional programming paradigms, offering flexibility in coding styles. Key features of JavaScript include its ability to manipulate the Document Object Model (DOM), allowing for dynamic manipulation of HTML and CSS elements on web pages. Additionally, JavaScript supports asynchronous programming through features such as callbacks, promises, and async/await, facilitating non-blocking I/O operations and enhancing performance. With the rise of frameworks and libraries such as React.js, AngularJS, and Vue.js, JavaScript has expanded its capabilities, enabling developers to build complex single-page applications (SPAs) and progressive web apps (PWAs). Beyond web development, JavaScript has also found applications in server-side programming (Node.js), game development, mobile app development (using frameworks like React Native and Ionic), and Internet of Things (IoT) development. Overall, JavaScript continues to evolve and innovate, playing a crucial role in shaping the modern web landscape.

**5.4 SAMPLE CODE**

<!DOCTYPE HTML>

<%@page import="java.io.File"%>

<%@page import="pack.mail"%>

<%@page import="cloudme.FileDownload"%>

<%@page import="pack.AppSecurity"%>

<%@page import="util.Constants"%>

<%@page import="java.nio.file.Paths"%>

<%@page import="java.nio.file.Files"%>

<%@page import="java.nio.file.Path"%>

<%@page import="java.io.FileOutputStream"%>

<%@page import="java.io.FileInputStream"%>

<%@page import="pack.DbConnector"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>

<html>

<head>

<title>Dual Access Control for Cloud-Based Data Storage and Sharing</title>

<meta name="description" content="website description" />

<meta name="keywords" content="website keywords, website keywords" />

<meta http-equiv="content-type"

content="text/html; charset=windows-1252" />

<link rel="stylesheet" type="text/css" href="style/style.css" />

</head>

<body>

<div id="main">

<div id="header">

<div id="logo">

<div id="logo\_text">

<br /> <br /> <font size="5"><a href="index.html"><span

class="logo\_colour">Dual Access Control for Cloud-Based Data Storage and Sharing

</div>

</div>

<div id="menubar">

<ul id="menu">

<li class="selected"><a href="userpage.jsp">Home</a></li>

<li><a href="logout.jsp">Logout</a></li>

</ul>

</div>

</div>

<div id="content\_header"></div>

<div id="site\_content">

<div id="sidebar\_container">

<br /> <br /> <br /> <br />

<div class="sidebar">

<div class="sidebar\_top"></div>

<div class="sidebar\_item">

<h3>Useful Links</h3>

<ul>

<li>

<a href="userpage.jsp">Home</a></li>

<li><a href="logout.jsp">Logout</a></li>

</ul>

</div>

<div class="sidebar\_base"></div>

</div>

</div>

<div id="content">

<div class="form\_settings">

<br /> <br /> <br />

<table style="width: 100%; border-spacing: 0;">

<tr>

<th>File Name</th>

<th>Get File Key</th>

<th>File Owner</th></tr>

if (view != null && view.equals("yes")) {

%>

<table style="width: 100%; border-spacing: 0;">

<tr>

<th>File Name</th>

<th><%=filename%></th>

</tr>

<tr>

<td align="center" colspan="2"><textarea rows="10"

cols="10" name="filedata"> <%=viewData%> </textarea></td>

</tr>

</table>

<%

}

}

}

%>

<%

}

%>

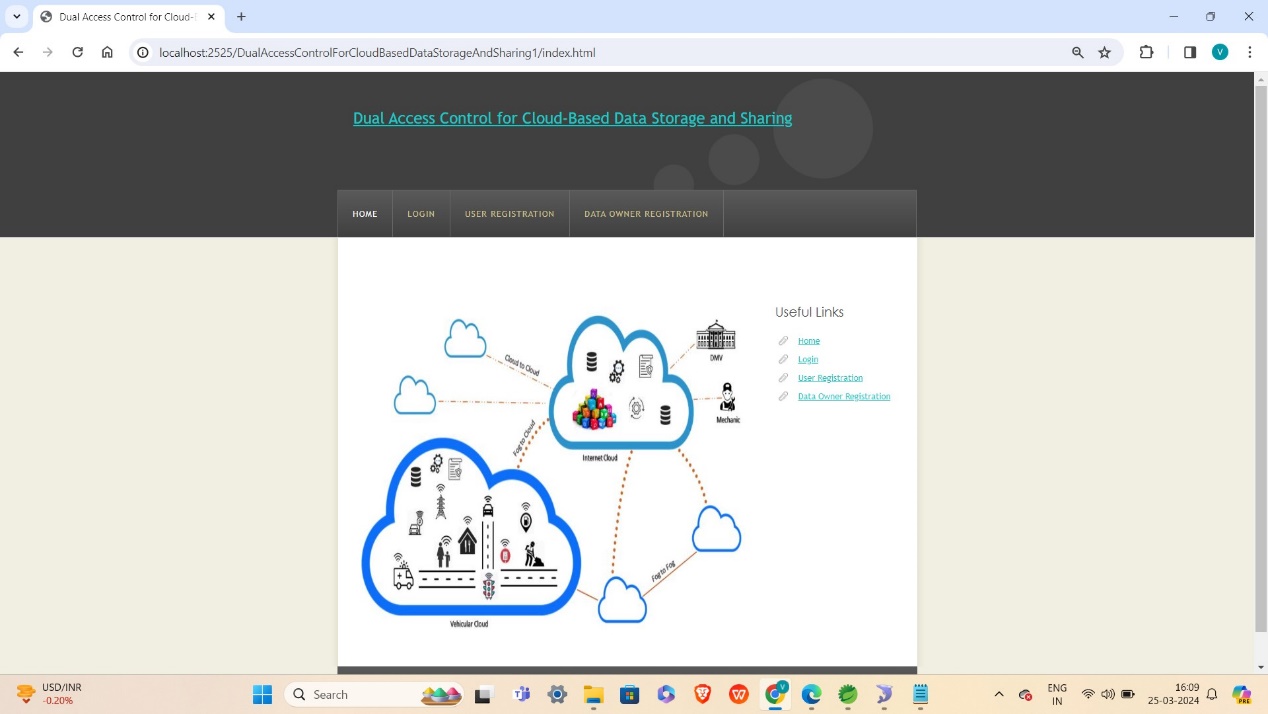
</div>

</div>

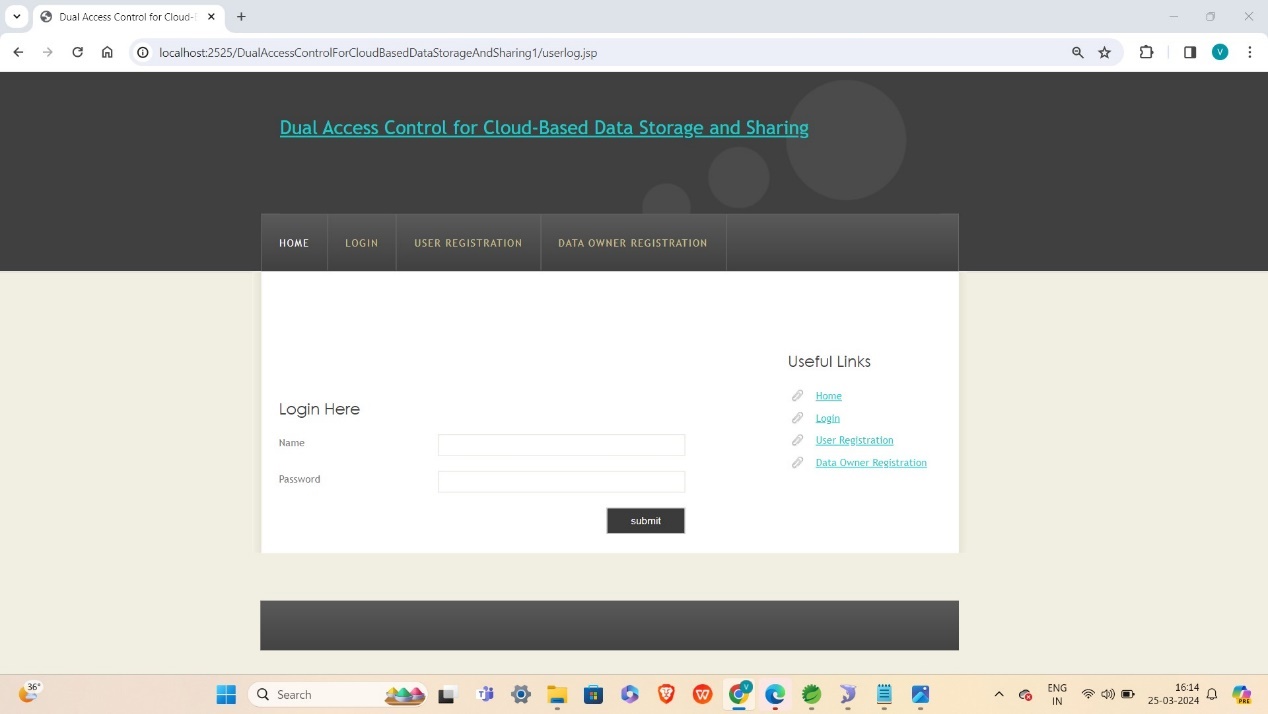
</div></body>

</html>

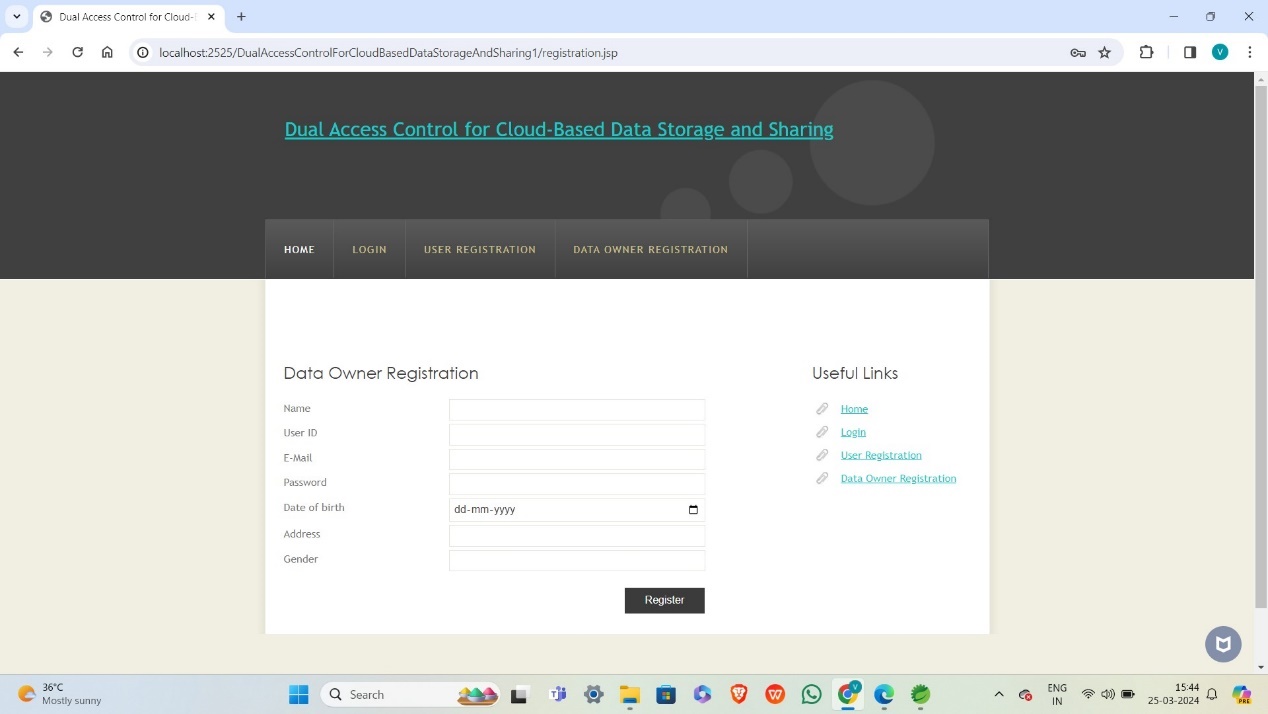
**6.EXPERMENT RESULTS**



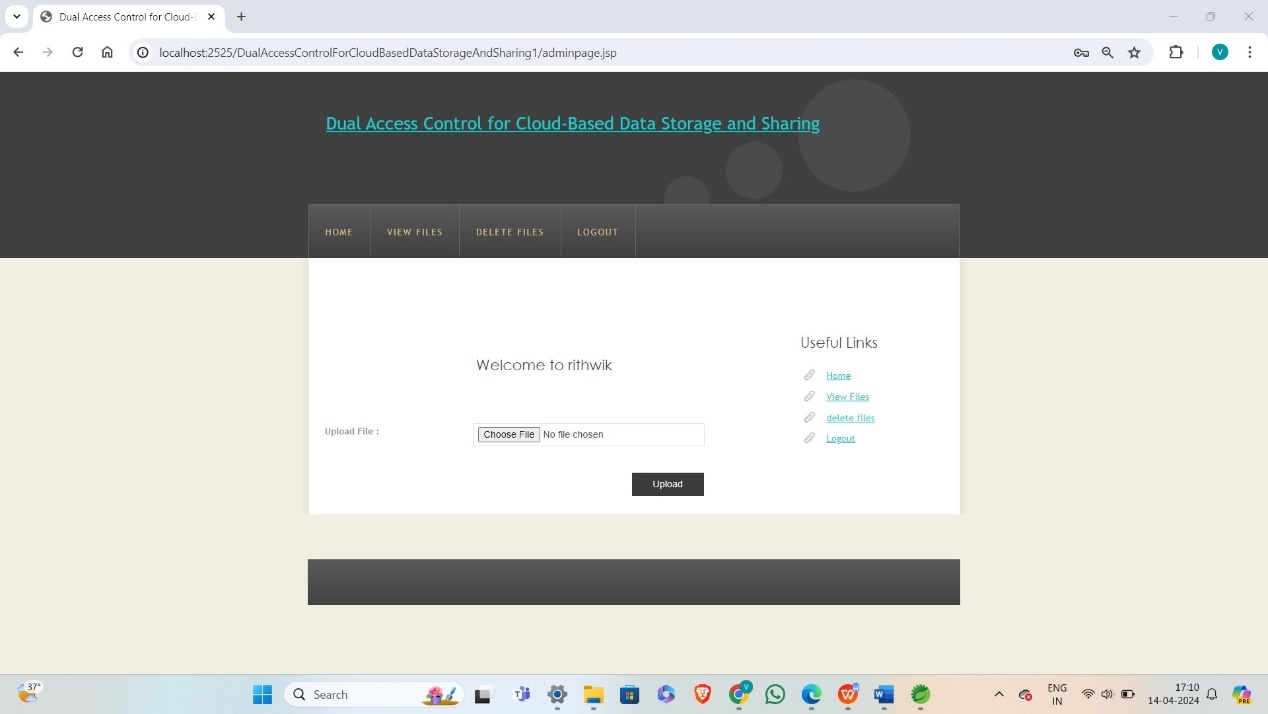
6.1 Home Screen



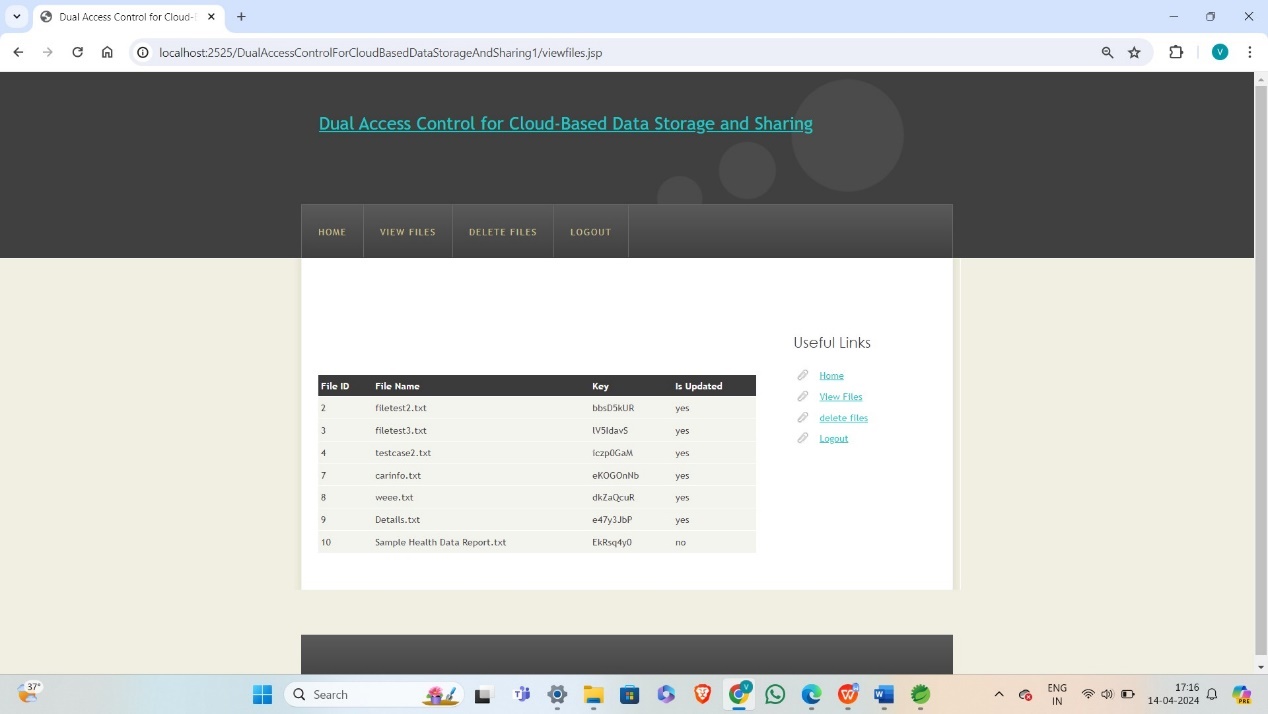
6.2 Login page



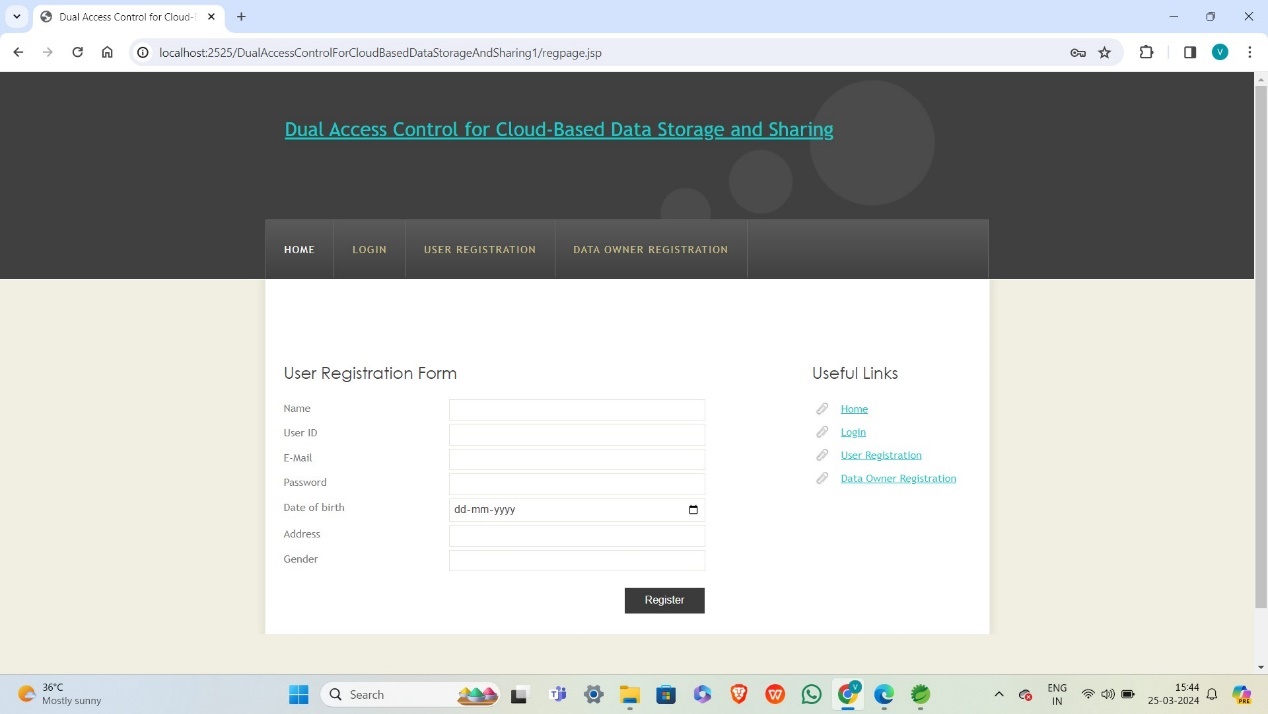
6.3 Owner Registration Page



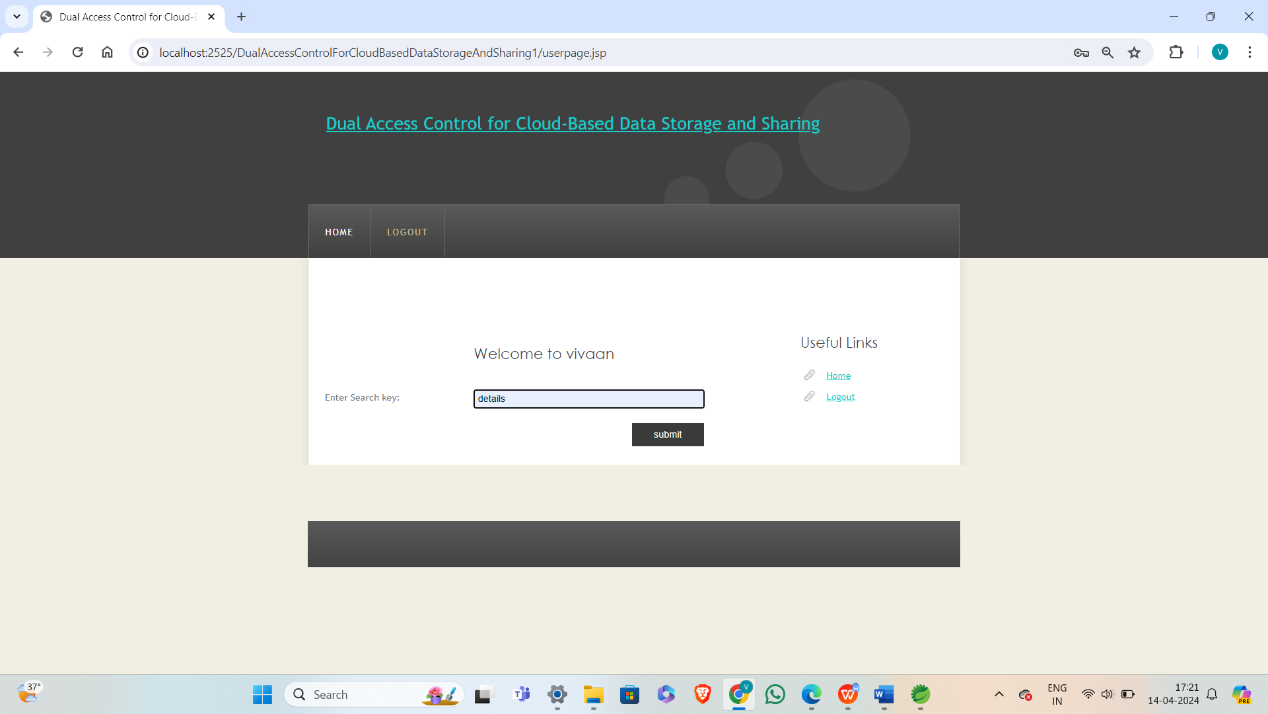
6.4 Owner file upload page



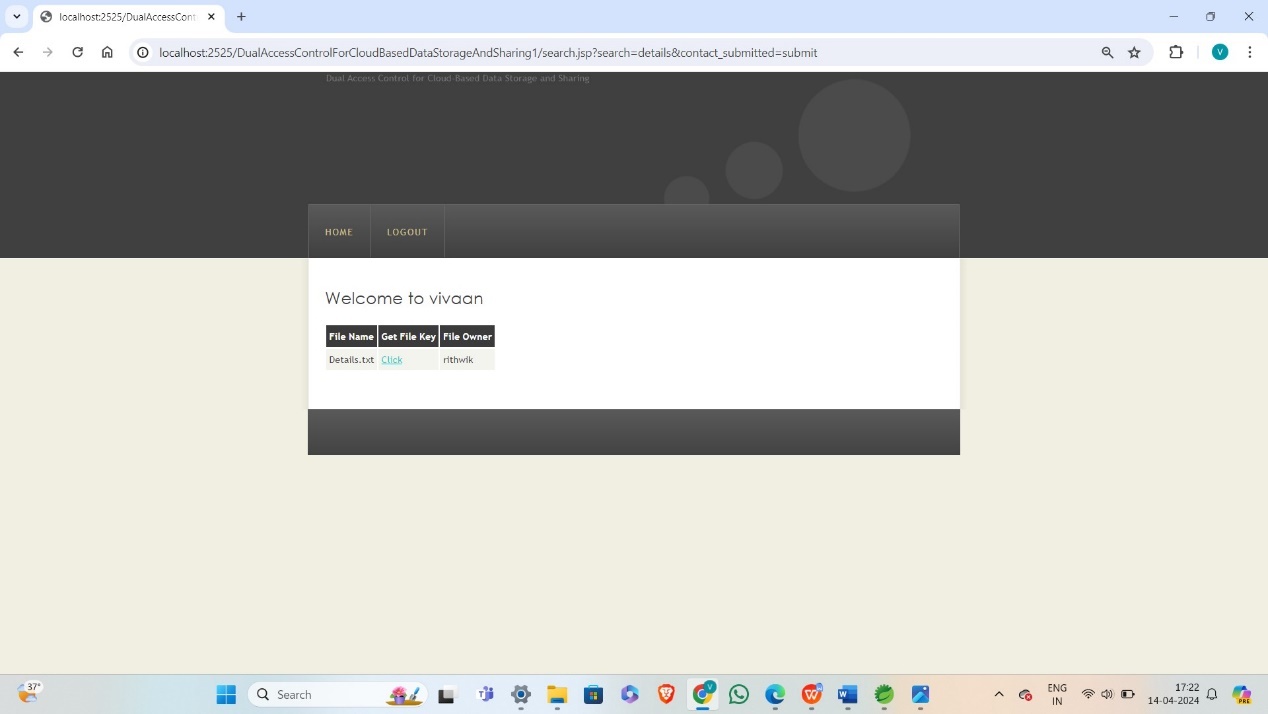
6.5 Owner uploaded files



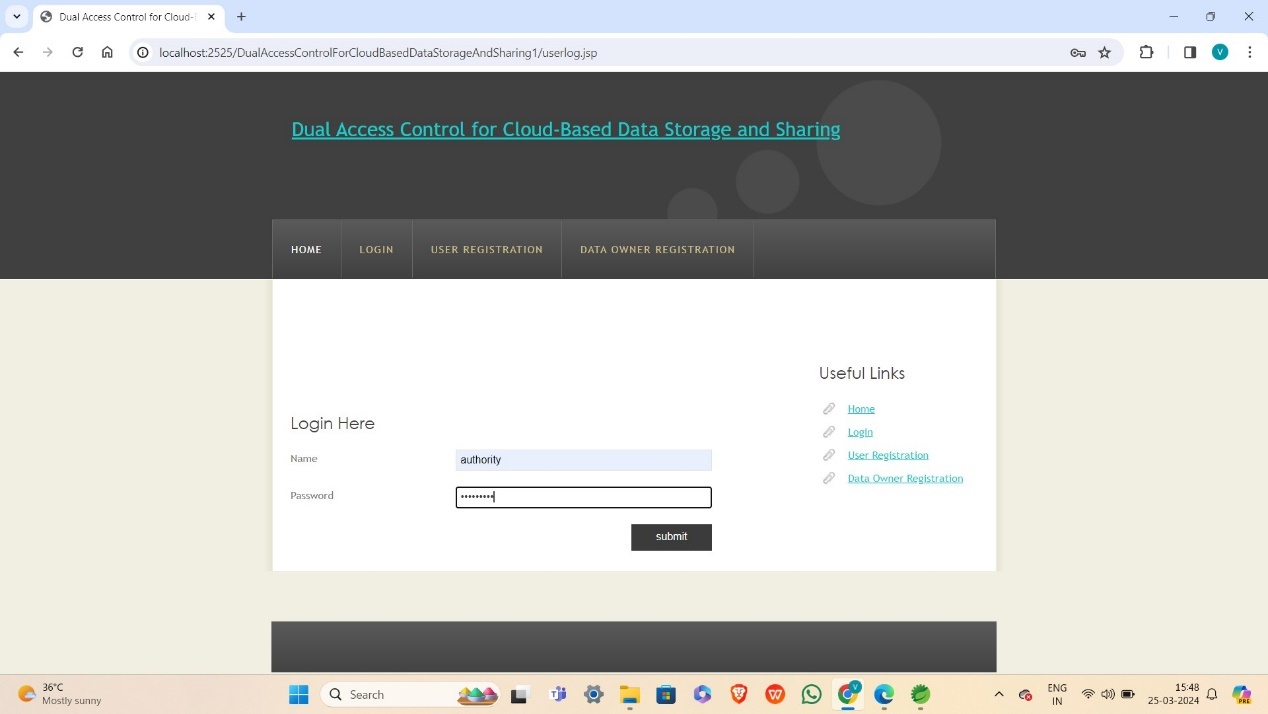
6.6 User Registration Page



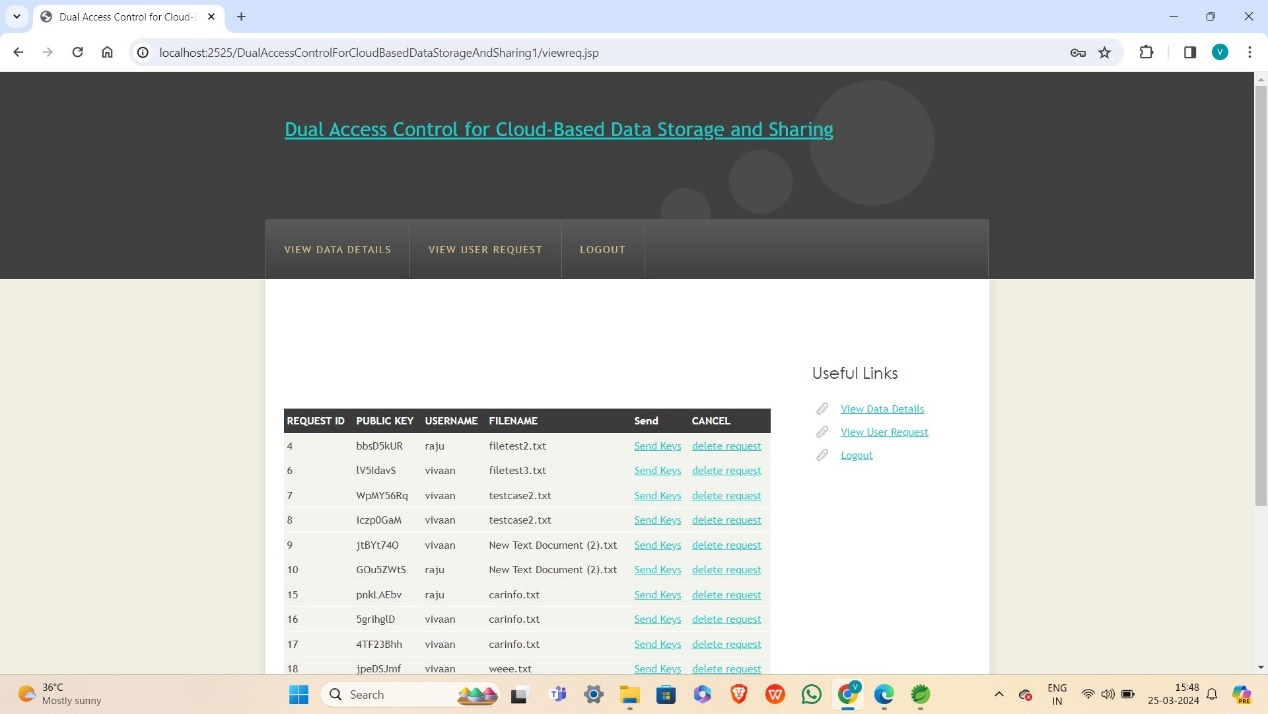
6.7 User searching file



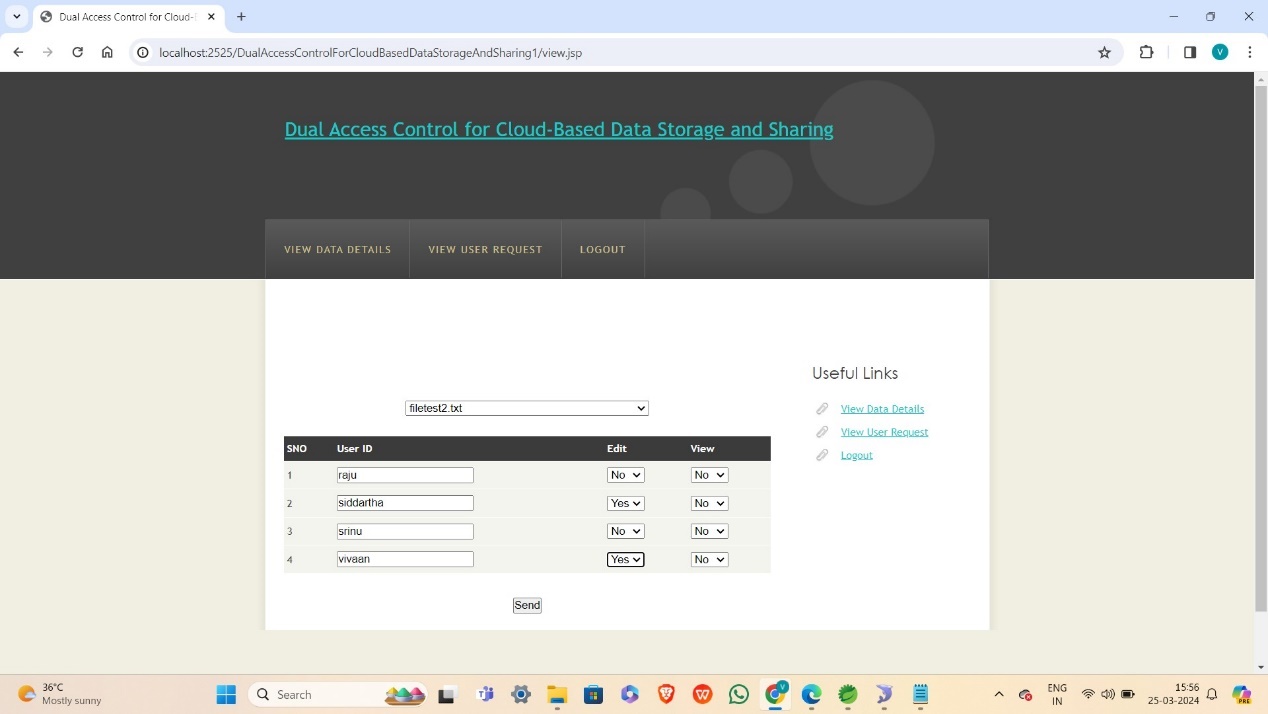
6.8 User requesting file key



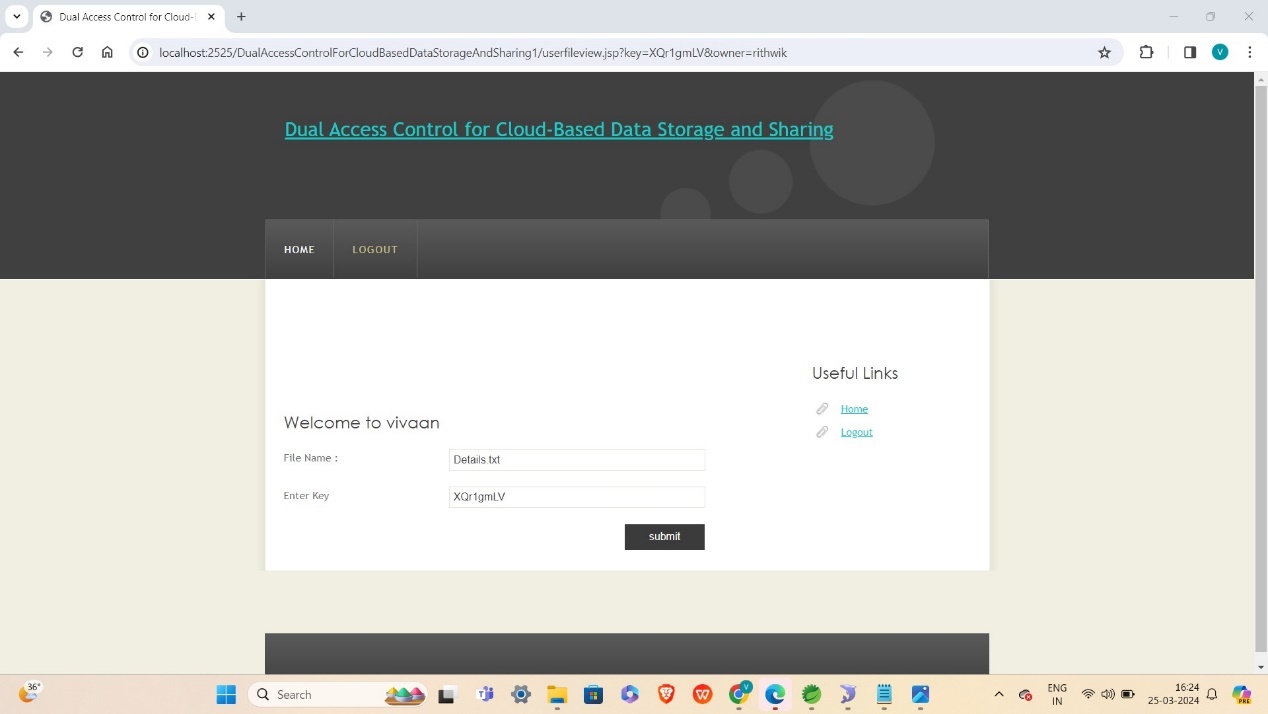
6.9 Authority Login



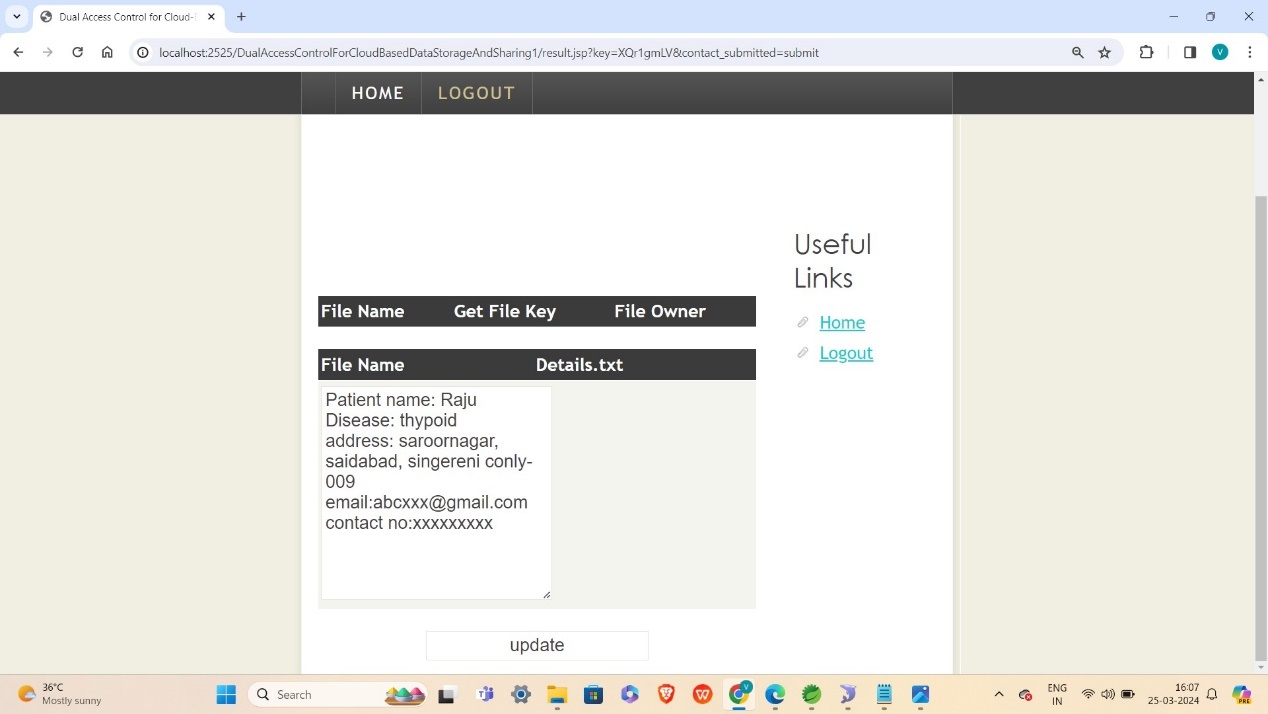
6.10 User Requested Files



6.11 Access control policy page



6.12 User Entering secret key



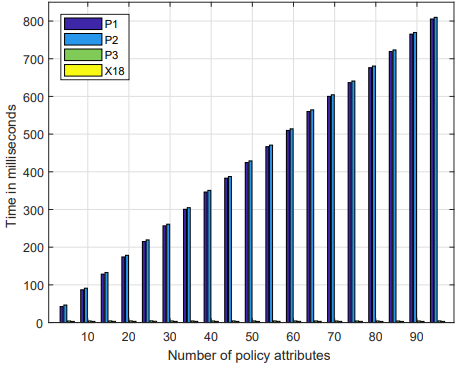
6.13 Requested File details

**6.15 Parameters with formulas:**

1.Risk Score Calculation: Risk Score = (Impact \* Likelihood) The risk score represents the potential impact of a security incident multiplied by the likelihood of its occurrence. Impact and likelihood are typically assigned numerical values based on risk assessments.2.Access Control Decision: Access Granted = (Authentication Level ≥ Authorization Level) This formula checks whether the authentication level (e.g., multi-factor authentication) of a user is equal to or higher than the authorization level (e.g., role-based access control) required to access a resource.3.Encryption Strength: Encryption Strength = Key Length × Key Strength The encryption strength is determined by the length of the encryption key (in bits) multiplied by the strength of the cryptographic algorithm used. Higher values indicate stronger encryption.4.Access Control Effectiveness: Access Control Effectiveness = (Number of Authorized Accesses / Total Access Attempts) × 100% This formula calculates the percentage of access attempts that are successfully authorized by the access control mechanisms. Higher values indicate more effective access control.

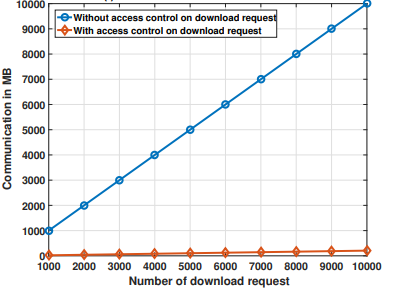
**7. Discussion Of Results**

**7.1 Experiment 1:** Comparison between the two proposed systems and other related work in terms of computational cost.

****

Graph 7.1.1

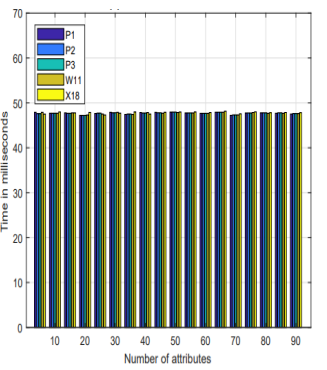
Findings: shows that the computational costs for shared files of our proposed systems are less than that of the strawman approach and the approach in . Fig. shows that the time cost for the procedure Download Request Generation of the basic system is the same with that of the enhanced system, and less than that of the strawman approach and the . Fig. illustrates that the time cost executed on the cloud side for the procedure Access Control on Download Request of the enhanced system is a little higher than that of the basic system.

**7.2 Experiment 2 :**Comparison between the two proposed systems and other related work in terms of communication cost. 

Graph 7.2.1

Findings:shows that the communication cost for the procedure Download Request Generation of the basic system is the same with that of the enhanced system, and less than that of the strawman approach and the approach in Fig. shows that our proposed systems dramatically reduce the communication overhead under EDoS attacks.

**7.3 Experiment 3:** Comparison between the two proposed systems and the CP-ABE system in terms of computational cost.



Graph 7.3.1

Findings:Fig. shows the experimental results in terms of computational cost. In particular, illustrate that the time costs for procedures Parameter Initialization, Data User Registration, Shared File Generation and Outsourcing, and Access Shared Data of the basic system and the enhanced system are the same (or almost the same) with that of the underlying CP-ABE .

**7.4 Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Test Name | Inputs | Process | Excepted Output | Actual Output | Status |
| 1 | Login Test | UserName, Password | Validate username and password in datatbase | Need to redirect to user home page | It’s Redirected to user home | Success |
| 2 | RegistrationTest | Username,  password, email,mobilenumber, email etc… | insert the users into database | Need to insert the user details into database | It’s inserted | Success |
| 3 | Upload file | File,Userid. | Insert the files into the cloud | Need to insert the files into the database | It’s inserted | Success |
| 4 | Grant access | Userid, Permission | Grant the access to the user to access the files | Need to grant permission to user | permission granted | Success |
| 5 | Send Mail | Userid, mailed, key, message | Send these inputs to specified user mail id | Need to send the details to mail id | Mail forwarded | success |
| 6 | View / Update | Userid, file id,key,permission | If user is having access to file and entered right file key ,then can view or update | update the file if given update option or else just view. | file updated | success |

**8.CONCLUSION**

We addressed an interesting and long-lasting problem in cloud-based data sharing, and presented two dual access control systems. The proposed systems are resistant to DDoS/EDoS attacks. We state that the technique used to achieve the feature of control on download request is “transplantable” to other CP-ABE constructions. Our experimental results show that the proposed systems do not impose any significant computational and communication overhead (compared to its underlying CP-ABE building block). In our enhanced system, we employ the fact that the secret information loaded into the enclave cannot be extracted. However, recent work shows that enclave may leak some amounts of its secret(s) to a malicious host through the memory access patterns or other related side-channel attacks . The model of transparent enclave execution is hence introduced in . Constructing a dual access control system for cloud data sharing from transparent enclave is an interesting problem. In our future work, we will consider the corresponding solution to the problem.

**9. FUTURE SCOPE**

The proposed dual access control system for cloud-based data storage presents numerous avenues for future development and exploration. Enhancements in security protocols, such as integrating advanced encryption techniques and multi-factor authentication methods, could further fortify data protection. Optimizing the system for scalability and performance under increasing data volumes and user loads is imperative, warranting ongoing research into algorithmic and architectural refinements. Adaptive access control policies that dynamically adjust based on contextual factors can improve responsiveness and resilience to evolving security threats. Integration with emerging technologies like blockchain for access log immutability or differential privacy techniques for preserving user privacy while analyzing data could expand the system's capabilities. Focusing on usability and user experience through user studies and feedback analysis can refine interface design and interaction mechanisms. Additionally, ensuring compliance with data privacy regulations and conducting real-world deployment studies with industry partners are crucial steps toward practical implementation and adoption. Overall, the future scope of this project encompasses a wide array of areas spanning security, performance, usability, compliance, and real-world applicability, offering ample opportunities for continued research and innovation.

**9.REFERENCES**

[1] Joseph A Akinyele, Christina Garman, Ian Miers, Matthew W Pagano, Michael Rushanan, Matthew Green, and Aviel D Rubin. Charm: a framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3(2):111–128, 2013.

[2] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.

[3] Alexandros Bakas and Antonis Michalas. Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm 2019, pages 472–486, 2019.

[4] Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[5] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE, 2007. [6] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptology ePrint Archive, 2016(086):1–118, 2016.

[7] Ben Fisch, Dhinakaran Vinayagamurthy, Dan Boneh, and Sergey Gorbunov. IRON: functional encryption using intel SGX. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 765–782, 2017.

[8] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Advances in Cryptology-CRYPTO 1999, pages 537–554. Springer, 1999.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In ACM CCS 2006, pages 89–98. ACM, 2006.

[10] Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Allen Au. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. IEEE transactions on information forensics and security, 10(3):665–678, 2015.

[11] Christofer Hoff. Cloud computing security: From ddos (distribut- ed denial of service) to edos (economic denial of sustainability). http://www. rationalsurvivability.com/blog/?p=66.

[12] Joseph Idziorek, Mark Tannian, and Doug Jacobson. Attribution of fraudulent resource consumption in the cloud. In IEEE CLOUD 2012, pages 99–106. IEEE, 2012.

[13] Simon Johnson, Vinnie Scarlata, Carlos Rozas, Ernie Brickell, and Frank Mckeen. Intel Rsoftware guard extensions: Epid provision-ing and attestation services. White Paper, 1:1–10, 2016.

[14] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. Inferring ﬁne-grained control ﬂow inside sgx enclaves with branch shadowing. In 26th USENIX Security Symposium, USENIX Security, pages 16–18, 2017.

[15] Jiguo Li, Xiaonan Lin, Yichen Zhang, and Jinguang Han. Ksf- oabe: outsourced attribute-based encryption with keyword search functionforcloudstorage. IEEETransactionsonServicesComputing, 10(5):715–725, 2017.

[16] Jiguo Li, Yao Wang, Yichen Zhang, and Jinguang Han. Ful- l veriﬁability for outsourced decryption in attribute based encryption. IEEE Transactions on Services Computing, DOI: 10.1109/TSC.2017.2710190, 2017.

[17] Wei Li, Kaiping Xue, Yingjie Xue, and Jianan Hong. Tmacs: A robust and veriﬁable threshold multi-authority access control system in public cloud storage. IEEE Transactions on parallel and distributed systems, 27(5):1484–1496, 2016.

[18] Ben Lynn et al. The pairing-based cryptography library. Internet: crypto. stanford. edu/pbc/[Mar. 27, 2013], 2006.