OM SAKTHI

DATA STORAGE AND PRIVACY PROTECTION USING AUTHENTICATED GROUP KEY TRANSFER PROTOCOL IN CLOUD COMPUTING

Main project submitted to the

Adhiparasakthi College of Arts and Science (Autonomous)

In partial fulfilment of the requirements for the award of degree

MASTER OF COMPUTER APPLICATIONS

Submitted By

G.SNEHA

(30122P08030)

Under the Guidance of

Mr.G.S.SENTHIL KUMAR M.C.A., M.E.,M.Phil.,

Assistant Professor

Department of Computer Science and Applications

ADHIPARASAKTHI COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)

G.B. NAGAR, KALAVAI-632 506

APRIL – 2024

OM SAKTHI

ADHIPARASAKTHI COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)
G.B. NAGAR, KALAVAI - 632506

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"DATA STORAGE AND PRIVACY PROTECTION USING AUTHENTICATED GROUP KEY TRANSFER PROTOCOL IN CLOUD COMPUTING** " was carried out by **G.SNEHA (30122P08030)** in partial fulfilment of the award the Degree of **MASTER OF COMPUTER APPLICATIONS**, during the **FOURTH** semester under the guidance of **Mr.G.S.SENTHIL KUMAR M.C.A.,M.E.,M.Phil., (Assistant Professor).**

SIGNATURE OF GUIDE **SIGNATURE OF HOD**

PRINCIPAL

Submitted for the fourth semester project work- viva voce held on_____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

MAXIMIZE
INFOTECH(P) LTD

Date: 13-March-2024

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Ms.Sneha.G** (Reg.no.: **30122P08030**), a student of **Adhiparasakthi College of Arts and Science,** has successfully completed her project titled "**DATA STORAGE AND PRIVACY PROTECTION USING AUTHENTICATED GROUP KEY TRANSFER PROTOCOL IN CLOUD COMPUTING"** on our company, with reference to the partial fulfillment of the requirements of the **MCA [Master of Computer Applications]** degree. This candidate has associated with us as a project trainee from 1st December 2023 to 3rd March 2024.

We wish her the very best in all her future endeavors. Thanking

you,

With Best Regards,

Kotteeswari M Resource Coordinator

No.1/A, 1st West Cross street, Opp. Govt B.Ed. and Law College,Gandhi Nagar, Katpadi, Tamil Nadu 632006.
Off: 0416-2241114, Cell: +91 9080183848,

TABLE OF CONTENTS

ABSTRACT

Group Key Management Protocol for file sharing on cloud storage (GKMP). Faced with network attacks from public channel, a group key generation scheme based on mixed encryption technology. And a verification scheme is used to prevent shared files from being attacked by the collusion attack of cloud providers' and group members'. Security and performance analyses indicate that the protocol is both secure and efficient for data sharing in cloud computing. Access control and group key management can be used for key management on file sharing. Key components of the project include the design and implementation of the Authenticated Group Key Transfer Protocol, integration with cloud storage systems, and the establishment of a secure and scalable framework. To secure group key management protocol on cloud storage over unreliable channels, aiming at protecting the shared les on the cloud storage. Mixed encryption technology is used to generate and distribute group keys, which resistance attacks from network monitor. In addition, we propose a verified protocol that against the attacks from the file sharers or the cloud provider. Aims to develop a robust system that ensures the confidentiality, integrity, and privacy of stored data. By employing an Authenticated Group Key Transfer Protocol, cryptographic keys are securely and efficiently distributed among designated groups. This process not only enhances data security but also enables controlled access to shared resources.

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along with the completion of my project. I that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank our Principal **Dr. A. MOHAMED SADIQ, M.Sc., M.Phil., Ph.D**., for encouragement to do the project in Maximize InfoTech and giving us all support and guidance. I am extremely thankful to him for providing such nice support and guidance, although he had a busy schedule managing the corporate affairs.

I heartily thank our HOD **Dr. P.V. PRAVEEN SUNDAR M.C.A., M.Phil., Ph.D**., for their encouragement and insightful suggestions over the completion of this project. I am sincerely thankful to him for providing such nice support and guidance.

I owe my deep gratitude to our project guide **Mr.G.S.SENTHIL KUMAR M.C.A.,M.E.,M.Phil.,**. Who took a keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I would not forget to remember our Department Staff for their encouragement and more over for their timely support and guidance till the completion of our project work. Also, I would like to extend our sincere esteems to all staff in the laboratory for their timely support.

I am extremely grateful to my Parents and My Brother for their love, prayers, caring and sacrifices for educating and preparing me for my future and My Special thanks goes to my friends for their support.

DATE: **G.SNEHA**

PLACE: KALAVAI **(30122P08030)**

LIST OF FIGURES

LIST OF TABLES

CHAPTER - I

INTRODUCTION

1.1 Background

Faced with today's innovative blow-up of cloud technologies, rebuilding services in terms of cloud have become more popular. In a shared-tenancy cloud computing environment, data from different clients which can be hosted on separate virtual machines may reside on a single physical machine.

In the dynamic landscape of cloud computing, where vast amounts of sensitive data are stored and processed, ensuring robust security measures and preserving user privacy have become paramount concerns. The growing adoption of cloud services has led to an increased risk of data breaches and unauthorized access, necessitating innovative solutions to fortify the storage infrastructure. One promising avenue involves the integration of group key management protocols.

This approach enhances the security of data storage by employing advanced encryption mechanisms, while simultaneously addressing privacy concerns through controlled access mechanisms. By implementing group key management protocols in cloud computing environments, organizations can strengthen their defenses against evolving cyber threats.

1.2 Objectives

- **Implement Group Key Management Protocols:**

  Develop and integrate advanced cryptographic protocols to manage group keys effectively within the cloud computing environment.

- **Enhance Data Security:**

  Strengthen data security measures through the application of encryption mechanisms, ensuring the confidentiality and integrity of stored information.

- **Controlled Access:**

  Implement access control mechanisms facilitated by group key management, enabling restricted and monitored access to sensitive data, thereby preventing unauthorized usage.

- **Preserve User Privacy:**

  Focus on designing and implementing privacy-preserving strategies,

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose:

The purpose of this project or research is to address the critical challenges associated with data storage and privacy protection in cloud computing environments. The focus is on implementing a robust solution that utilizes an authentication group key transfer protocol. The key objectives include:

- Enhancing the security of data stored in cloud environments.
- Providing a reliable authentication mechanism to control access to sensitive information.
- Implementing a group key transfer protocol to ensure secure communication within authorized groups.
- Mitigating potential privacy risks associated with cloud storage solutions.

1.3.2 Scope:

The scope of this research revolves around advancing the security infrastructure of cloud computing through the implementation of an authentication group key transfer protocol. With a primary focus on data storage and privacy protection, this project aims to develop a comprehensive solution that ensures the confidentiality and integrity of stored data.

The scope encompasses the creation of a robust authentication mechanism, designed to verify user identities accessing cloud resources. Additionally, the project delves into the formulation and implementation of a group key transfer protocol, facilitating secure communication within authorized user groups. The research spans various aspects of cloud environments, addressing challenges associated with data storage security, user authentication, and privacy preservation.

1.3.3 Applicability:

This project's findings and solutions are applicable to a range of scenarios within the broader context of cloud computing. Applicability extends to:

- **Enterprises and Organizations:**

  Businesses leveraging cloud services for data storage and processing.

- **Government Institutions:**

  Public sector entities relying on cloud solutions for efficient data management.

- **Cloud Service Providers:**

  Companies providing cloud services, enhancing their security measures.

- **Research and Development:**

  Academia and research institutions working on advancements in cloud security.

1.4 Achievements

- Successful development and implementation of an authentication group key transfer protocol for bolstering the security infrastructure in cloud computing.
- Ensuring heightened assurance of data confidentiality and integrity through the secure group key transfer protocol.
- Implementation of an efficient and reliable authentication mechanism for streamlined user verification in cloud environments.
- Achievement in establishing secure communication within authorized user groups, safeguarding sensitive data from unauthorized access.
- Creation of a versatile solution applicable to diverse cloud computing architectures and platforms.
- Successful integration of measures to protect user privacy, mitigating risks associated with unauthorized access and data breaches.
- Demonstrating relevance across various sectors, including enterprises, government institutions, and cloud service providers.
- Addressing legal and regulatory aspects related to data privacy, ensuring compliance with relevant laws and standards.
- Balancing security measures to enhance the overall user experience without compromising usability.
- Recognition and potential adoption of the developed protocol by cloud service providers, enterprises, or institutions seeking advanced data storage and privacy protection solutions.

1.5 Organization of report

Title Page:

- Title of the Report: "Data Storage and Privacy Protection in Cloud Computing using Authentication Group Key Transfer Protocol"

Abstract:

- Concise summary of the research, including objectives, methodology, key findings, and implications.

Table of Contents:

- Clearly list the sections and subsections with corresponding page numbers.

Introduction:

- Background and context of the research.
- Problem statement and motivation.
- Objectives and scope of the study.
- Overview of the authentication group key transfer protocol.
- Explanation of the proposed system

Literature Review:

- Review of existing literature related to data storage, privacy protection, and authentication mechanisms in cloud computing.
- Identification of gaps in the current state of research.
- Review of key technologies, protocols, and frameworks relevant to the project.

Methodology:

- Description of the research methodology.
- Data collection methods.
- Tools and technologies used.

System Architecture:

- Illustration and explanation of the proposed system architecture.
- Components of the system and their interactions.
- Diagrams, charts, or graphs to aid understanding.

Implementation:

- Detailed account of the implementation process.
- Challenges encountered and solutions adopted.

Results:

- Presentation of research findings.
- Data analysis, including quantitative and qualitative results.
- Visualization of results using graphs or charts.

Discussion:

- Interpretation of results in the context of research objectives.
- Comparison with existing literature.
- Implications of the findings for data storage and privacy protection in cloud computing.

Conclusion:

- Summary of key findings.
- Contributions of the research.
- Limitations and areas for future research.

References:

- Comprehensive list of all sources cited in the report.
- Recognition of individuals or organizations that contributed to the project.

CHAPTER – II

SURVEY OF TECHNOLOGIES

2.1 Java Technology

Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer.

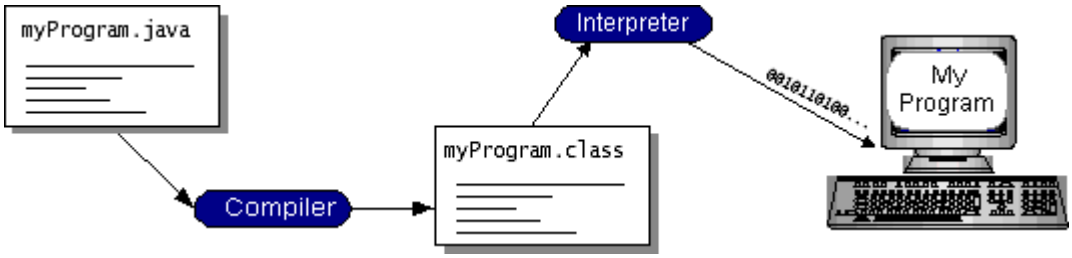

Fig 2.1 Process

The Java Platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of otherhardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

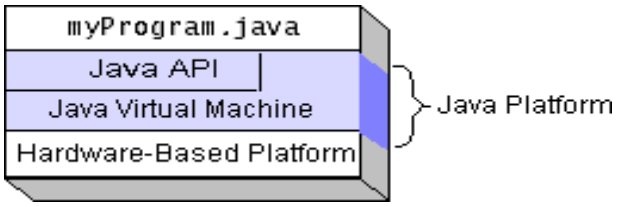

Fig 2.2 Java Platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers

What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components**: Known as JavaBeans™, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™)**: Provides uniform access to a wide range of relational databases.

2.2 Introduction to MySQL Server

MySQL is a database server

MySQL is ideal for both small and large applications

MySQL supports standard SQL

MySQL compiles on a number of platforms

MySQL is free to download and use

PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software and MySQL AB.

- MySQL is a database management system.

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.

- MySQL is a relational database management system.

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. MySQL software is Open Source.

History

We started out with the intention of using the MySQL database system to connect to our tables using our own fast low-level (ISAM) routines. However, after some testing, we came to the conclusion that MySQL was not fast enough or flexible enough for our needs. This resulted in a new SQL interface to our database but with almost the same API interface as mSQL. This API was designed to allow third-party code that was written for use with mSQL to be ported easily for use with MySQL.

The derivation of the name MySQL is not clear. Our base directory and a large number of our libraries and tools have had the prefix "my" for well over 10 years. However, co-founder Monty Widenius's daughter is also named My. Which of the two gave its name to MySQL is still a mystery, even for us.

The name of the MySQL Dolphin (our logo) is "Sakila," which was chosen by the founders of MySQL AB from a huge list of names suggested by users in our "Name the Dolphin" contest. The winning name was submitted by Ambrose Twebaze, an Open Source software developer from Swaziland, Africa. According to Ambrose, the feminine name Sakila has its roots in SiSwati, the local language of Swaziland. Sakila is also the name of a town in Arusha, Tanzania, near Ambrose's country of origin, Uganda.

Main features

The following list describes some of the important characteristics of the MySQL Database Software. See also Section 1.6, "MySQL Development Roadmap", for more information about current and upcoming features.

Internals and Portability:

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms, "Operating Systems Supported by MySQL".
- Uses GNU Automake, Autoconf, and Libtool for portability.
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, APIs and Libraries.
- Fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available.
- Provides transactional and non-transactional storage engines.
- Uses very fast B-tree disk tables (MyISAM) with index compression.
- A very fast thread-based memory allocation system.

CHAPTER-III

REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

In the contemporary landscape of cloud computing, the escalating reliance on cloud- based data storage presents a pressing challenge regarding the security and privacy of stored data. The imperative to safeguard sensitive information necessitates a comprehensive solution that integrates an authenticated transfer protocol. This protocol plays a pivotal role in ensuring the authenticity and integrity of data during both storage and transmission processes. The defined problem encompasses multifaceted issues such as preventing unauthorized access, fortifying defenses against potential data breaches, and establishing robust user authentication mechanisms. The challenge lies in developing and implementing a robust solution that leverages an authenticated transfer protocol to ensure the privacy and integrity of data during storage and transmission. This involves addressing specific issues such as unauthorized access, data breaches, user authentication, and compliance with data protection regulations.

Moreover, compliance with stringent data protection regulations adds a layer of complexity, demanding solutions that adhere to legal frameworks. As cloud environments handleincreasingly vast volumes of data, scalability and performance considerations become paramount. The proposed solution must seamlessly integrate with existing cloud storage systems while upholding the highest standards of data privacy and security. In essence, the challenge at hand involves developing an encompassing framework that addresses the intricate interplay of data storage, privacy protection..

3.2 Requirements Specification

The requirements specification for data storage and privacy protection using an authenticated transfer protocol in cloud computing is designed to establish a comprehensive framework for addressing the critical challenges in this domain. The system's functional requirements necessitate a secure and scalable data storage mechanism within cloud environments, ensuring data redundancy and incorporating reliable backup mechanisms.

The solution's scalability and performance requirements should be articulated to handle the increasing volumes of data typical in cloud environments. Seamless integration with existing cloud storage systems is also vital. In essence, the comprehensive requirements encompass data storage, secure transfer protocols, user access controls, regulatory compliance, and system scalability, collectively forming the foundation for a robust and privacy-centric solution in cloud computing.

This protocol should guarantee the integrity and authenticity of data during transmission, thereby mitigating the risks associated with unauthorized access and data breaches. The system is expected to seamlessly integrate with existing cloud storage infrastructures while adhering to the highest standards of data protection regulations. Overall, this specification aims to guide the development of a solution that not only meets the immediate requirements of secure data storage and privacy protection but also anticipates future scalability and compliance needs within the dynamic landscape of cloud computing.

3.3 Planning and Scheduling

The planning process involves determining the key tasks required to achieve the objectives, and putting them together to create a work plan or schedule. It is probably advisable notto plan far-off tasks in too much detail, but to think in terms of a number of project stages, often referred to as project milestones. During the operational stages of the project, the various milestoneswill be planned in greater detail. However it is important to determine the critical path, those activities which are dependent on one another. Planning also involves allocating resources to the various tasks, be they people, equipment, facilities, outside suppliers, etc.

All this information is put together in a project plan, which is a major control document for the project, to be constantly referred to and updated.

Phase	No. of iterations	Start	End
Inception phase	1	Week1(28/12/23)	Week3(14/01/24)
Elaboration phase	2	Week3 (15/01/24)	Week6 (15/02/24)
Construction phase	3	Week6 (16/02/24)	Week9 (03/03/24)
Transition phase	1	Week9 (04/03/24)	Week12 (29/03/24)

Table 3.1 Project plan

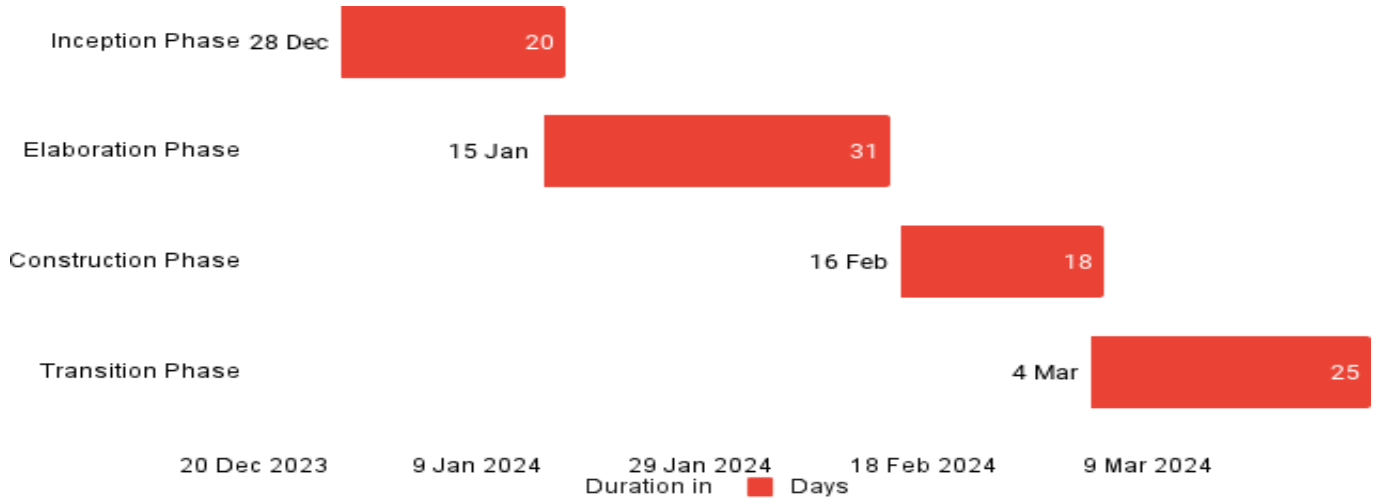

Fig 3.1 Gantt chart for Project Plan

Task name	Start	Finish
Architectural Prototype (Design)	28/12/23	01/01/24
Construction (Coding, Testing)	16/02/24	03/03/24
Transition (Implementation and Training)	04/03/24	29/03/24

Table 3.2 Project schedule

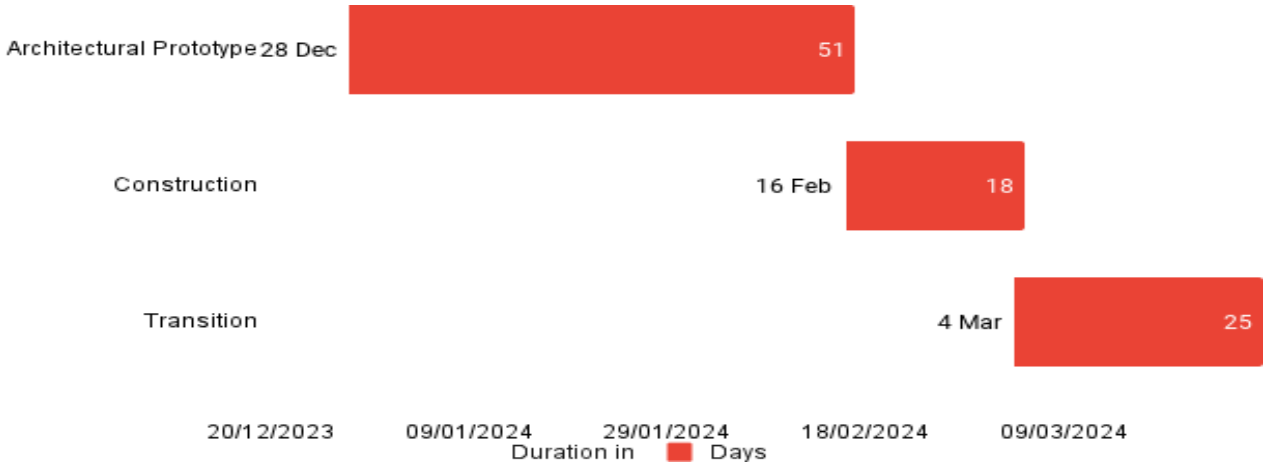

Fig 3.2 Gantt chart for Project Schedule

3.4 Software and Hardware Requirements

HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 1 TB
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- RAM : 8 GB

SOFTWARE REQUIREMENTS:

- Operating system : Windows 7/10.
- Coding Language : Java
- Tool : NetBeans 7.2.1
- Database : MYSQL

3.5 Preliminary Product Description

The proposed product aims to address the burgeoning challenges associated with data storage and privacy protection in the realm of cloud computing by employing an advanced authenticated transfer protocol. This comprehensive solution seeks to ensure the secure and scalable storage of data in cloud environments while prioritizing the confidentiality and integrity of the information stored. Leveraging cutting-edge encryption techniques and secure communication channels.

Vision and Impact:

The system's primary features include a resilient and scalable cloud-based data storage infrastructure, designed to handle increasing data volumes while minimizing the risk of data loss through redundancy and efficient backup mechanisms. Furthermore, the authenticated transfer protocol will authenticate the entities involved in data transactions, mitigating the potential threats of unauthorized access and data breaches.

Seamless Integration:

User authentication and authorization mechanisms will be integral components, ensuring that access to sensitive data is tightly controlled, with configurable privileges based on user roles. The product will adhere to stringent regulatory requirements for data protection, providing organizations with a compliance-oriented solution.

Data Protection Regulations:

Emphasizing ease of integration, the product is designed to seamlessly integrate with existing cloud storage systems, facilitating a smooth transition for organizations already invested in cloud computing. The envisioned solution aims to set a new standard for data security in the cloud, offering a holistic approach that combines robust data storage practices.

User Access Controls:

User Authentication:

- Explain the system's user authentication mechanisms to verify identity of users interacting with the stored data.

Authorization Framework:

- Detail how the product controls access privileges.

3.6 Conceptual Models

An Entity-Relationship (ER) model for cloud computing typically focuses on representing entities and the relationships between them within the cloud computing context. Here's a simplified ER model for cloud computing:

Entities:

- User:

  Attributes: UserID (Primary Key), Username, Email, Password, Role

- Cloud Provider:

  Attributes: ProviderID (Primary Key), ProviderName, ServicesOffered, Location

- Data Storage:

  Attributes: StorageID (Primary Key), StorageType, Capacity, EncryptionStatus

- Virtual Machine (VM):

  Attributes: VMID (Primary Key), VMType, CPU, RAM, StorageAllocation

Relationships:

- Utilizes:

  Connects User with Cloud Provider, indicating the cloud provider utilized by each user.

  Relationship Attributes: UsageStartDate, UsageEndDate

- Stores:

  Associates Cloud Provider with Data Storage, representing the storage resources offered by a cloud provider.

  Relationship Attributes: StorageStartDate, StorageEndDate

- Uses:

  Connects User with Authenticated Transfer Protocol, representing the protocols utilized by each user for data transfer.

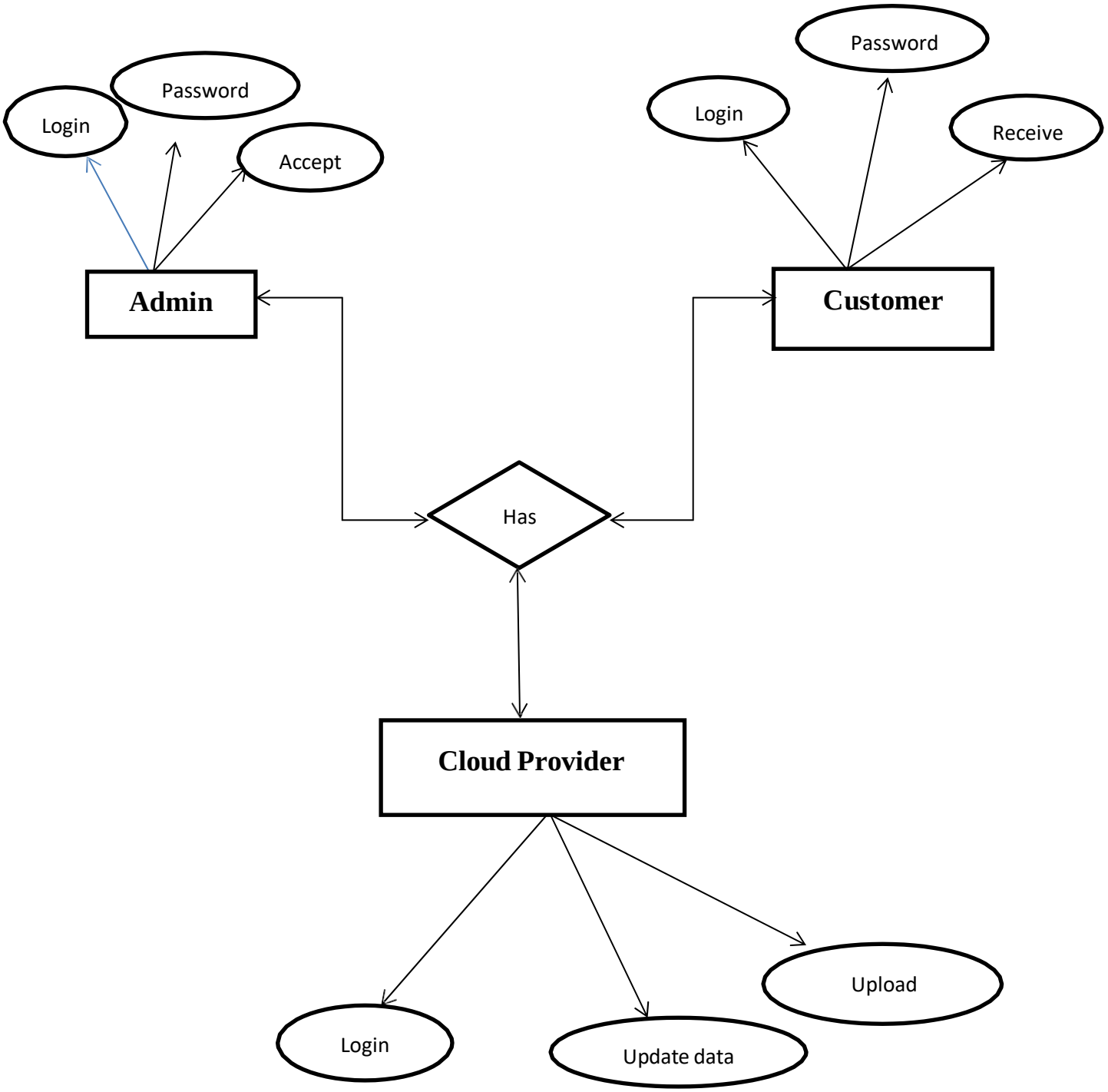

Fig 3.3 ER Diagram

CHAPTER – IV

SYSTEM DESIGN

4.1 Basic Modules

A module is a logically separable part of a program. It is a program unit that is discrete and identifiable with respect to compiling and loading. Partitioning a system in two modules is useful only if the modules are solvable and modifiable. Separately a system is considered modular if it consists of discrete components such that each component supports a well-defined abstraction, and if a change to one component has a minimal impact on other component. Brief description of each of the two modules is given below. To develop a basic module for data storage and privacy protection in cloud computing, employing an authenticated group key transfer protocol, several fundamental steps must be undertaken. Initially, the system architecture needs to be carefully designed, delineating the cloud storage provider, Data encryption plays a pivotal role in this module, requiring the implementation of robust encryption algorithms like AES before storing data in the cloud. Authentication and authorization protocols must be integrated to ascertain that only authenticated users can access the stored data, employing techniques such as OAuth or JWT.

- Cloud provider Module
- Data Owner Module

4.1.1 Cloud Provider Module

Define the components of your system, including the cloud storage provider, client applications, and any intermediary services. Determine how data will be encrypted, stored, and accessed securely. Audit trails and logging functionalities are indispensable for monitoring data access and key management activities, aiding in compliance adherence with relevant regulations like GDPR or HIPAA.

Choose a suitable protocol for securely transferring keys among authorized users. Protocols like Secure Multi-Party Computation (SMPC), Secure Group Communication (SGC), or any other authenticated group key transfer protocol can be considered.

4.1.2 Data Owner Module

Generate and manage encryption keys securely. Implement mechanisms for key distribution, rotation, and revocation as per the selected protocol**.**

Authentication and Authorization:

- Implement user authentication mechanisms to ensure that only authorized users can access the data.

- Use techniques like OAuth, JWT (JSON Web Tokens), or other authentication protocols.

Secure Communication Channels:

- Ensure secure communication channels between clients and the cloud service to prevent eavesdropping or tampering.
- Use HTTPS/TLS for securing communication over the network.

Privacy Protection Measures:

- Implement techniques such as differential privacy, data anonymization, or data masking to protect sensitive information.
- Define access control policies to restrict data access based on user roles and permissions.

Audit Trails and Logging:

- Maintain logs of all data access and key management activities for auditing purposes.
- Implement mechanisms to detect and respond to security incidents.

Testing and Validation:

- Thoroughly test the implemented module to ensure its correctness and security.
- Perform penetration testing and vulnerability assessments to identify and mitigate potential security risks.

Compliance and Regulations:

- Ensure compliance with relevant data protection regulations such as GDPR (General Data Protection Regulation), HIPAA (Health Insurance Portability and Accountability Act), etc.

Documentation and Training:

- Document the system architecture, security measures, and procedures for key management and data access.
- Provide training to users and administrators on how to use the system securely.

Continuous Monitoring and Maintenance:

- Establish a process for continuous monitoring of the system for security threats and vulnerabilities.
- Regularly update and patch system components to address known security issues.

By following these steps, you can develop a basic module for data storage and privacy protection using an authenticated group key transfer protocol in cloud computing. It's important to keep in mind that security is an ongoing process, and it requires constant vigilance and updates to adapt to evolving threats and technologies.

Audit trails and logging functionalities are indispensable for monitoring data access and key management activities, aiding in compliance adherence with relevant regulations like GDPR or HIPAA. Rigorous testing, validation, documentation.

4.2 Data Design

4.2.1 Schema Design

When designing the schema for data storage and privacy protection using an authenticated group key transfer protocol in cloud computing, it's essential to consider various aspects such as data encryption, access control, key management, and auditing. This schema design allows for the storage of encrypted data, management of encryption keys, and definition of access control policies, auditing of user activities, and implementation of authenticated group key transfer protocols. When designing the schema for data storage and privacy protection using an authenticated group key transfer protocol in cloud computing, it's essential to consider various aspects such as data encryption, access control, key management, and auditing. It provides a foundation for building a secure and privacy-protective data storage system in cloud computing environments. on specific requirements and considerations of the organization. Below is a simplified schema design:

User Table:

This table stores information about users accessing the system.

Fields: UserID (Primary Key), Username, Password (Hashed), Email, Role, etc.

Data Table:

This table stores the encrypted data along with metadata.

Fields: DataID (Primary Key), EncryptedData, DataOwnerID (Foreign Key), Timestamp, etc.

Key Table:

This table manages encryption keys for data access.

Fields: KeyID (Primary Key), DataID (Foreign Key), Key, KeyOwnerID (Foreign Key), Timestamp, etc.

Group Table:

This table defines groups of users for group-based access control.

Fields: GroupID (Primary Key), GroupName, Description, etc.

User_Group Table:

This table establishes the relationship between users and groups.

Fields: UserGroupID (Primary Key), UserID (Foreign Key), GroupID (Foreign Key), etc.

Access Control Table:

This table defines access control policies for data.

Fields: AccessControlID (Primary Key), DataID (Foreign Key), UserID (Foreign Key), AccessLevel, etc.

4.2.2 TABLE DESIGN:

Fields:

DataID (Primary Key): Unique identifier for each piece of data.

EncryptedData: Field to store the encrypted data.

Table 4.1 Admin

S.NO	FIELD NAME	NULL	TYPE	CONSTRAINT
1.	Username	Null	Text	-
2.	Password	Null	Text	-

Table 4.2 Data Owner

S.NO	FIELD NAME	NULL	TYPE	CONSTRAINT
1.	Name	Not Null	int(11)	Primary Key
2.	Email_id	Null	Text	-
3.	Password	Default Null	Text	-

Table 4.3 Group_Member

S.NO	FIELD NAME	NULL	TYPE	CONSTRAINT
1.	Name	Null	Text	-
2.	Email	Null	Text	-
3.	Select Group	Null	Text	-
4.	Password	Null	Text	-

Table 4.4 Cloud Provider

S.NO	FIELD NAME	NULL	TYPE	CONSTRAINT
1.	Id	Null	Text	-
2.	Name	Null	Text	-
3.	Email_id	Null	Text	-
4.	Password	Null	Text	-
5.	OTP	Null	Text	-

4.2.3 Data Integrity and Constraints

Constraints are the rules enforced on the data columns of a table. These are used to limit the type of datathat can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints could be either on a column level or a table level. The column level constraints are appliedonly to one column, whereas the table level constraints are applied to the whole table.

Integrity constraints are used to ensure accuracy and consistency of the data in relational database.Data integrity is handled in a relational database through the concept of referential integrity.

There are many types of integrity constraints that play a role in Referential Integrity (RI). These constraints include Primary Key, Foreign Key, Unique Constraints and other constraints.

In this Project, we use following constraints to maintain data integrity

- ✓ NOT NULL Constraint − Ensures that a column cannot have NULL value.
- ✓ UNIQUE Constraint − Ensures that all values in a column are different.
- ✓ PRIMARY Key − Uniquely identifies each row/record in a database table.
- ✓ FOREIGN Key − Uniquely identifies a row/record in any of the given database table.

4.3 Procedural Design

Procedural design for data storage and privacy protection in cloud computing, employing an authenticated group key transfer protocol, necessitates a systematic approach to ensure robust security measures are in place. Initially, data classification is crucial, discerning sensitive data that requires heightened protection. Selection of a suitable authenticated group key transfer protocol, such as Diffie- Hellman or Group Domain of Interpretation (GDOI), follows.

Establishing the cloud infrastructure involves implementing security measures like firewalls, intrusion detection, and encryption mechanisms. A tailored encryption mechanism, encompassing both symmetric and asymmetric algorithms, is designed to safeguard data during transit and at rest. Access control policies, enforced through role-based or attribute-based mechanisms, regulate data access.

Continuous monitoring and logging facilitate the detection of anomalous activities, while regular audits and compliance checks ensure adherence to industry standards and regulations. User training and awareness programs fortify security measures against human errors, and ongoing evaluation and improvement of security measures enable adaptation to emerging threats and technological advancements. This comprehensive procedural design ensures the effective storage and protection of data in cloud computing environments while upholding privacy and security through authenticated group key transfer protocols.

4.3.1 Logic Diagrams

In conceptualizing the procedural design for data storage and privacy protection utilizing an authenticated group key transfer protocol in cloud computing, a logical diagram serves as a foundational tool for visualizing the interconnected components and their operational relationships. The logic diagram outlines the sequential flow of activities, starting with data classification to identify sensitive information, followed by the selection and implementation of an appropriate authenticated group key transfer protocol.

- ➢ Use Case Diagram
- ➢ Activity Diagram

Use Case Components

Actor and use case are the main components of the use case diagram which are used to explain the access of function for a particular participant in an application. It seems like you're asking about the actor component within the context of use case components.

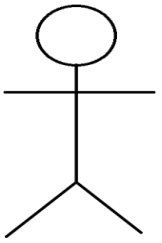

Fig 4.1 Actor Component

The symbol mention in the fig 4.2 is used to represent the user of the function or use case of the project. It contains a label below which helps us to identifier the user category. When discussing the actor component within a use case component, it typically refers to identifying and defining the different types of users or external systems that interact with the system being developed.

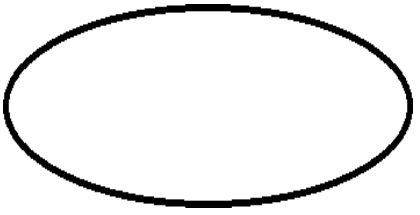

Fig 4.2 Use case Component

The functions of the project are representing using an ellipse with a label inside them which mention that name of the functionality. The "relation component" within the context of use case components typically refers to the relationships between different use cases or between use cases and actors in a system.

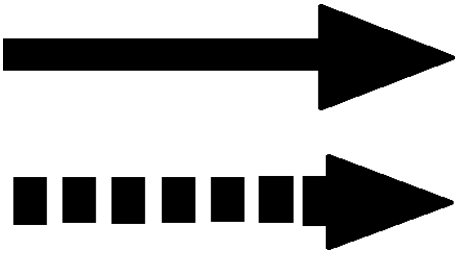

Fig 4.3 Relationship Component

4.3.1.1 Use Case Diagram

The use case diagram delineates the interactions between various actors, such as users, administrators, and external systems, and the system itself. For instance, actors may include data owners, cloud service providers, and regulatory bodies, each with distinct roles and responsibilities within the system. Use cases represent specific tasks or actions that these actors undertake to interact.

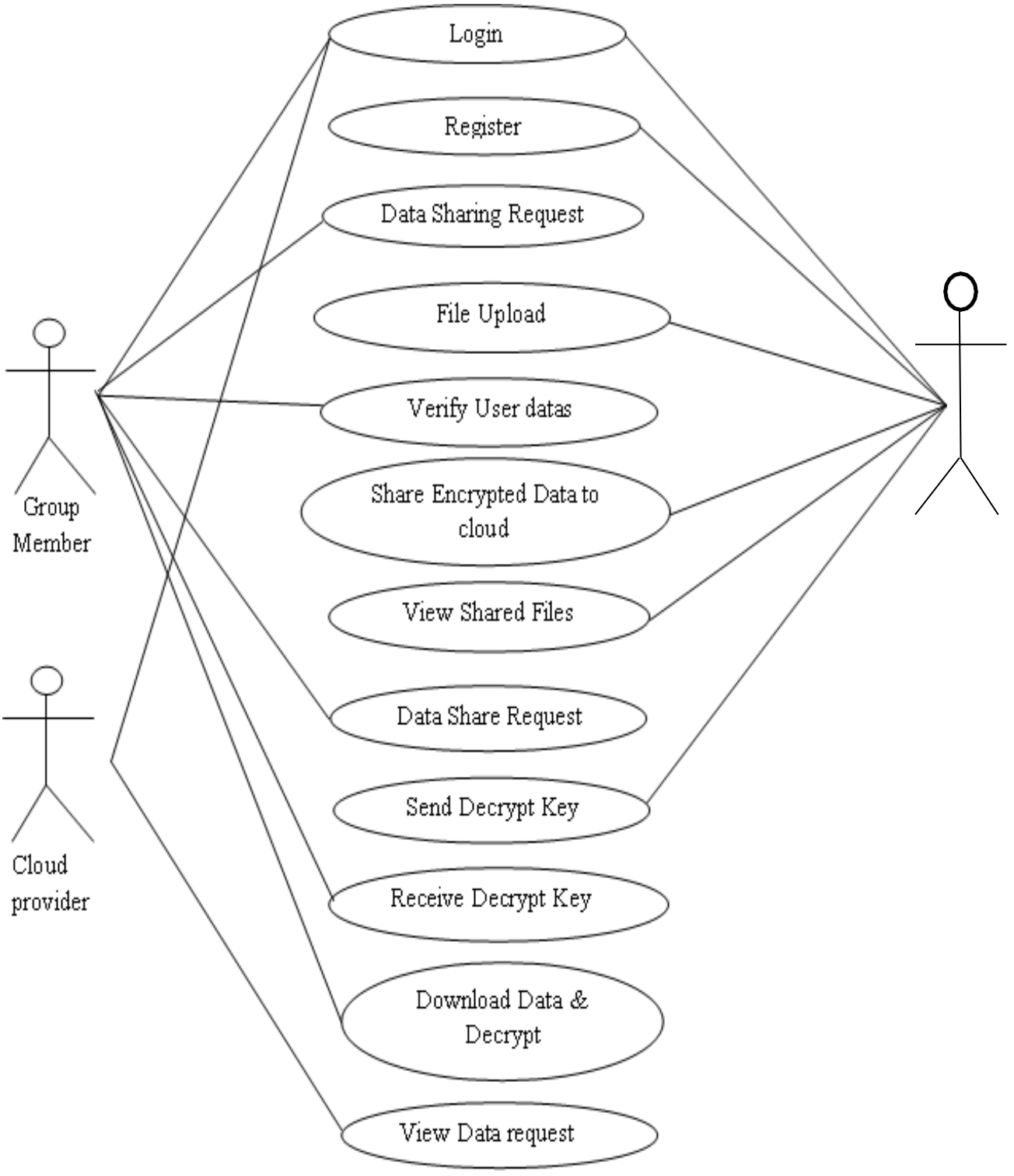

Fig 4.4 Use Case Diagram

4.3.1.2 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.

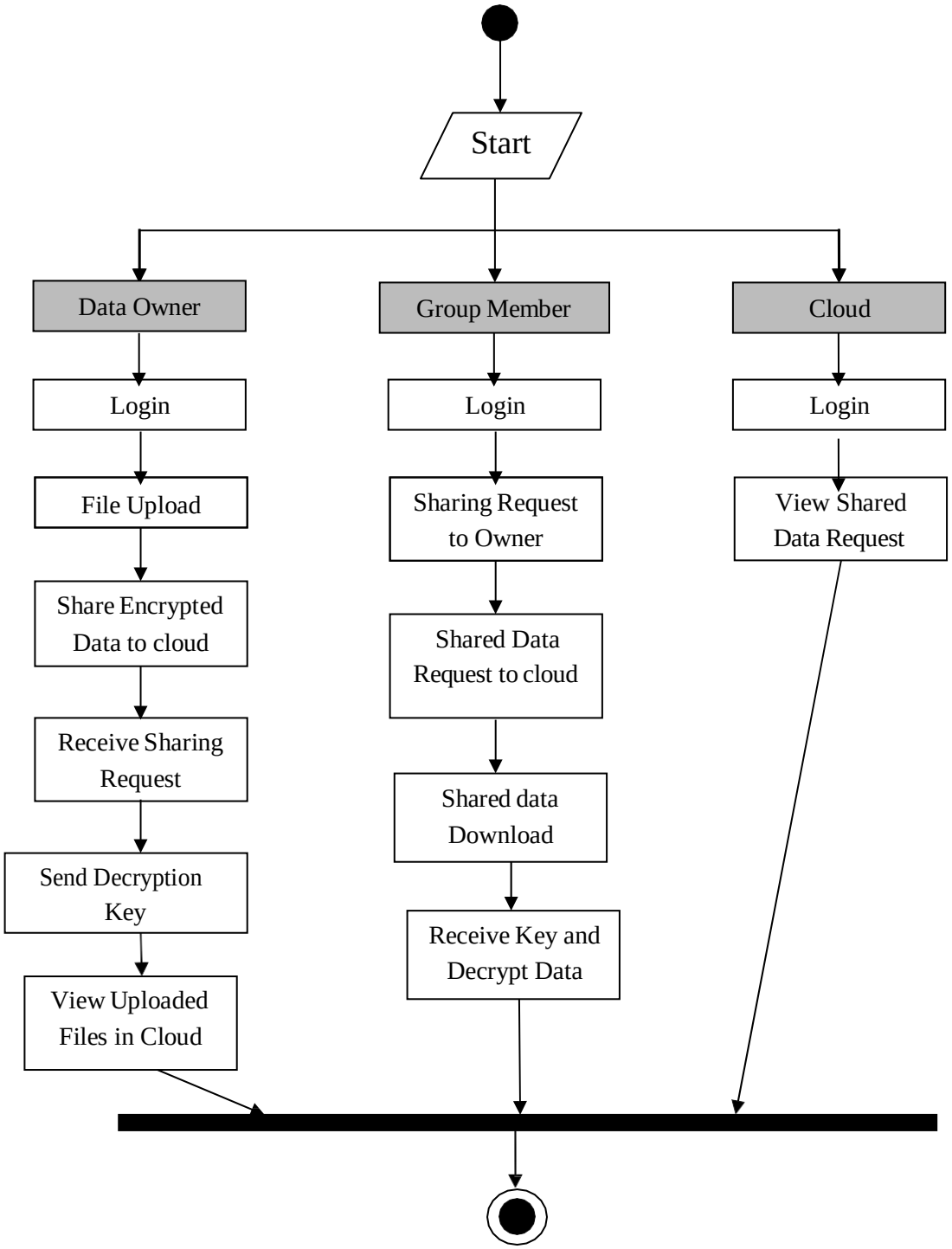

Fig 4.5 Activity Diagram

4.4 SECURITY ISSUES

A security issue refers to a vulnerability, flaw, or weakness within a system, process, or environment that has the potential to be exploited, leading to unauthorized access, disclosure, alteration, or destruction of sensitive information, disruption of services, or other forms of harm. Security concerns in cloud computing are paramount due to the shared nature of resources, distributed architecture, and reliance on third-party providers. One significant issue revolves around data privacy and confidentiality. Since data is stored and processed on remote servers, there's a risk of unauthorized access or data breaches, especially if encryption and access controls are not appropriately implemented. Additionally, data location and jurisdictional issues can arise, raising concerns about compliance with data protection regulations.

User authentication

User authentication is a process that allows a system to verify the identity of someone who accesses the application resource. Without User authentication, the system will be accessible to every so there is a chance of malfunction. So the project implementation must require the User authentication process to enhance the security of the projects. To enable the user authentication process user requested to enter their username and password to enter in to the application through which our system identifies the user.

User authorization

Authorization is the function of providing different access to differently based on their roles and responsibilities. In user authentication the system check only the validity of user but in user authorization system provides limited access based on the user type after identifying them using user authentication. So the system requires both user authentication and user authorization for securing the application.

SQL injection

SQL injection is a code injection technique, used to attack data-driven applications, in which various SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker) SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL injection attacks allow attackers to spoof identity.

Voiding transactions

Tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

Effectiveness of crime analysis effort

One of the critical security issues in crime analysis systems lies in the potential for data breaches and unauthorized access. These systems often handle highly sensitive information related to ongoing investigations, individuals involved in criminal activities, and even victims or witnesses.

4.5 Test Case design

Test case design is a critical phase in ensuring the reliability and functionality of a system, especially one as important as crime analysis with intrusion detection. In developing test cases for a crime analysis and intrusion detection system, it's imperative to cover a diverse range of scenarios to validate its effectiveness. Start with positive test cases, where the system is expected to behave as intended. This includes scenarios like successful login by authorized personnel, accurate incident detection, and appropriate generation of alerts. Negative test cases are equally important, focusing on scenarios where the system should identify and handle exceptions or anomalies. This might involve testing for incorrect login credentials, simulated intrusion attempts, or unexpected data inputs.

Table 4.5 Test case

Project: Data Storage and Privacy protection using Authenticated group key transfer protocol in Cloud Computing			**Module**: Data Owner		
Functionality:	To View cloud provider and group member details				
Pre-Condition:	Run the Application				
Test Case No#	1				
Scenario:	Adding Group member details in to database				
No#	Procedure	Test Condition	Test Data	Expected Result	Status
1	**6.** Run the application and load the login page.	Enter the username and password & click on login button Other wise	Username and Password	User form will be displayed Error Message Displayed.	Success Failed
2	1. Run the Application and load the home page. 2. Click on the "View Group member " menu and Select Area Details Tab	Detector have to fill all necessary fields with proper data Other wise	Nil	Cloud Details will be added to Database Error Message will be displayed	Success Failed
3	**6.** Run the Application and load the home page.	Detector have to fill all necessary fields with proper data	Nil	Group member and cloud provider Details are included in Database	Success

CHAPTER – V

IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. An "implementation approach" refers to the strategy or methodology used to translate a conceptual design or plan into a tangible and functional system or product. In the context of software development or system design, an implementation approach outlines how the various components, features, and functionalities will be developed, integrated, and deployed to create the final product.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Functional tests provide a systematic demonstration that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered 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.

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.

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration- oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.2 CODING DETAILS AND CODE EFFICIENCY

The coding details and efficiency phase describes how the coding performance changes the implementation of various task each of which performs a specific task.

5.2.1 Code Efficiency

Code efficiency plays a significant role in applications in a high execution-speed environment where performance and scalability are paramount.

Recommendations for code efficiency include:

- To remove unnecessary code or code that goes to redundant processing
- To make use of optimal memory and non-volatile storage.
- To ensure the best speed or run time for completing the algorithm
- To make use of error and expectation handling of all layers of software, such as the user interface, logic, and data flow
- To create programming code that ensures data integrity and consistency
- To develop programming code that's compliant with the design logic and flow
- To use the best keywords, data types and variables, and other available programming concepts to implement the related algorithm.

5.3 Testing Approaches

Testing is an important phase of the software Developing Phase. Testing represents an interesting anomaly for the software. Thus, a series of testing is performed for the system before the system is ready for user acceptance testing. The common view of testing held by users is that, it is performed to prove that there are no errors in the program. A Program is bug-free only till the next bug is found. The intention of testing is to find the 'next bug'. As an additional benefit, testing demonstrates.

5.3.1 Unit Testing

It is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed description as a guide, important paths are tested to uncover errors within the boundary of the modules.

- All independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute all error handling paths.

5.3.2 Validation Testing

It is the step where requirements established as a part of the software requirements analysis are validated against the software meet all functional behavioral performance requirements and the errors which are uncovered during the testing are correct. Form level and as well as field-level validations are performed in all the data entry screens.

5.3.3 Integration Testing

Modules are put together one by one, and the interfaces of modules are tested. In this project, it is tested in such a way that the session variables can be shared anywhere in the project and with the actual value. It is tested so that the flow from one form to another form is not affected due to the session variables.

5.3.4 Program Testing

It is nothing but a number of programs that form a cluster to achieve a certain goal. During program testing, two kinds of errors will occur namely, syntax errors and logical errors. Syntax errors have to be corrected before the program is executed. Logical errors may occur due to the incorrect handling of data, improper sequence of program statements, etc.

5.3.5 White Box Testing

White box testing is a testing technique that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing. White Box Testing Techniques includes

- **Statement Coverage**: This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage:** This technique is running a series of tests to ensure that all branches are tested at least once
- **Path Coverage:** This technique corresponds to testing all possible paths which means that each statement.

5.3.6 Black Box Testing.

It is sometimes called behavioral testing or partition testing. It attempts to find errors in the following categories.

- Incorrect or Missing functions.
- Interface errors.
- Errors in Data Structures or external database access.
- Performance errors and termination errors.

CHAPTER – 6

RESULTS AND DISCUSSION

6.1 TEST REPORTS

Fig 6.1 HOME PAGE

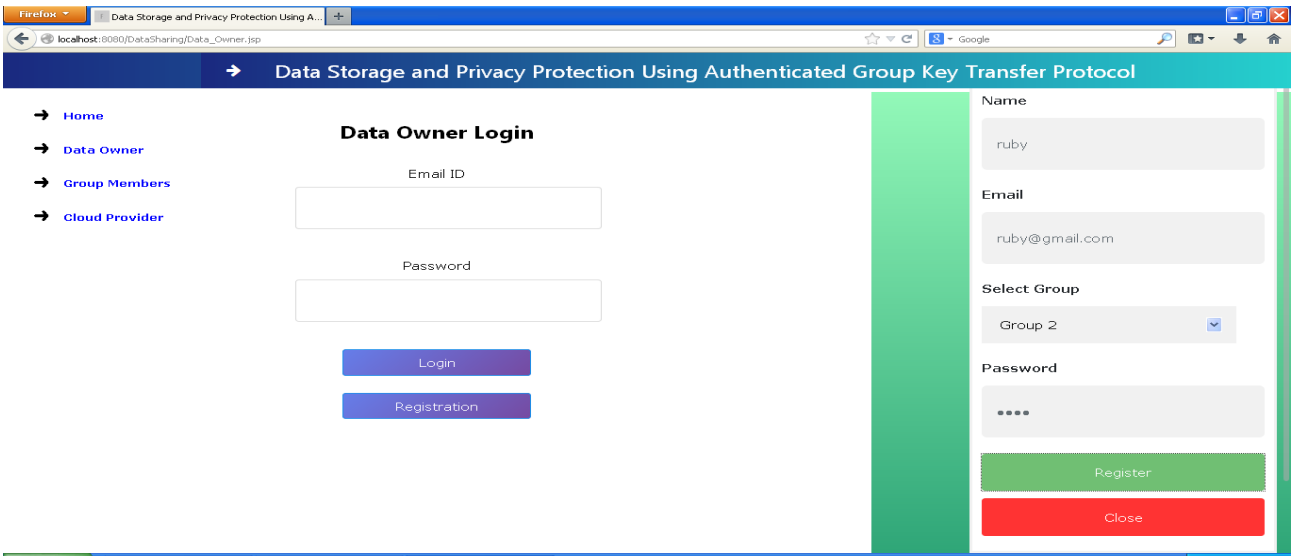

Fig 6.2 DATA OWNER REGISTERS

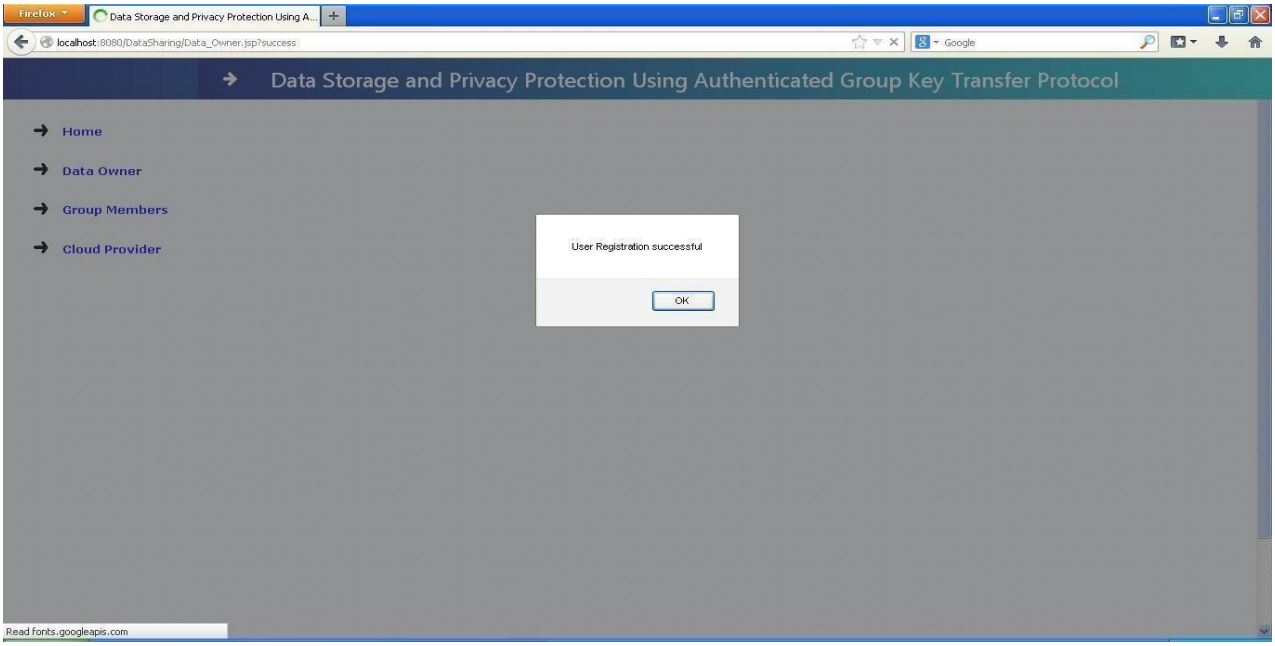

Fig 6.3 USER SUCCESSFUL REGISTRATION

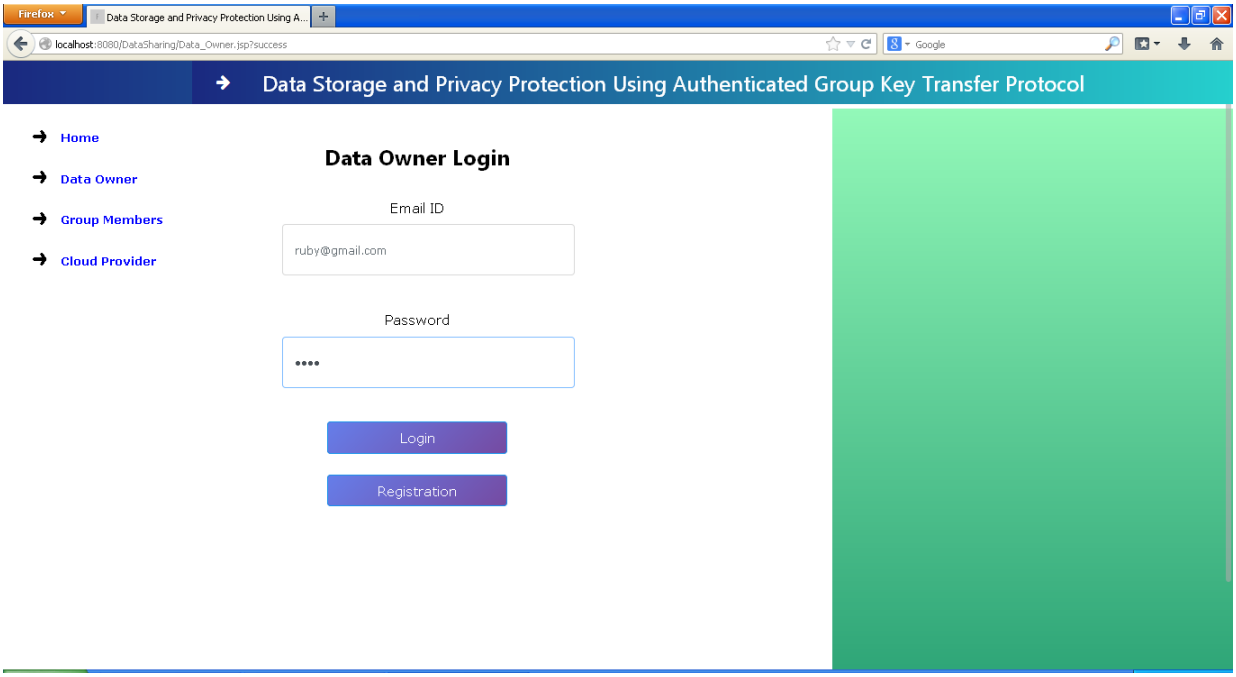

Fig 6.4 DATA OWNER LOGIN

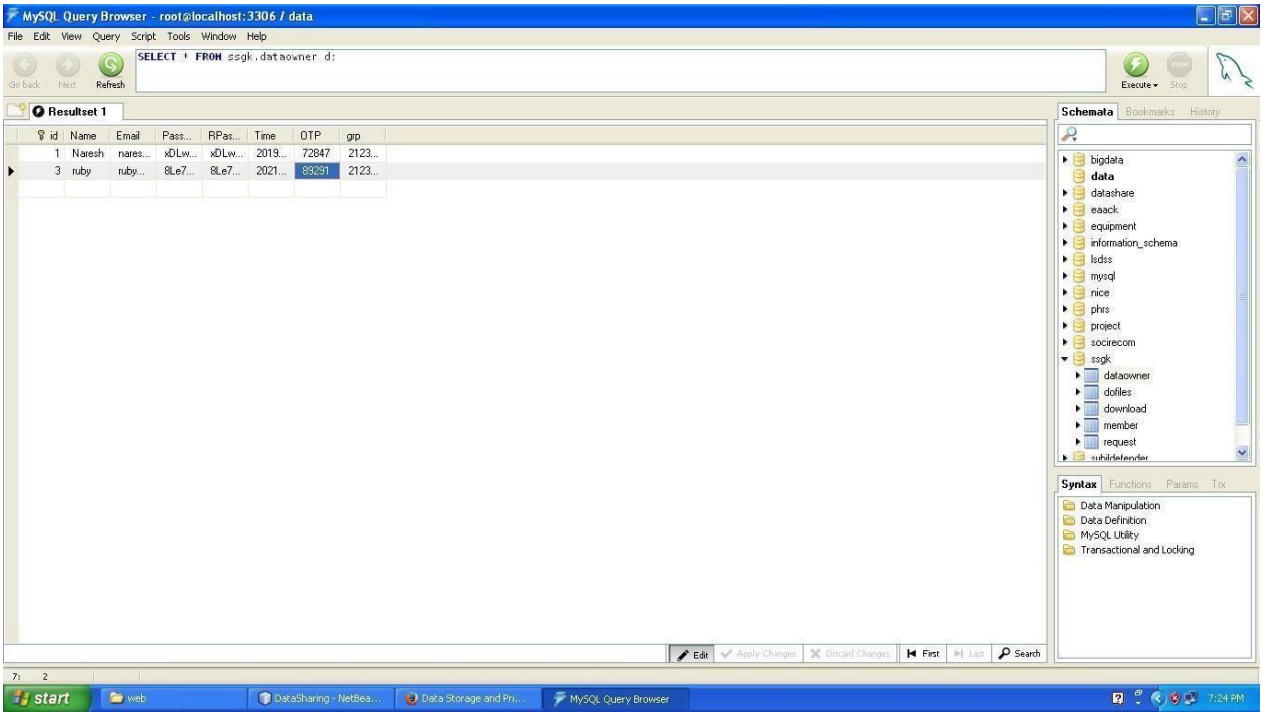

Fig 6.5 OWNER DATA PAGE

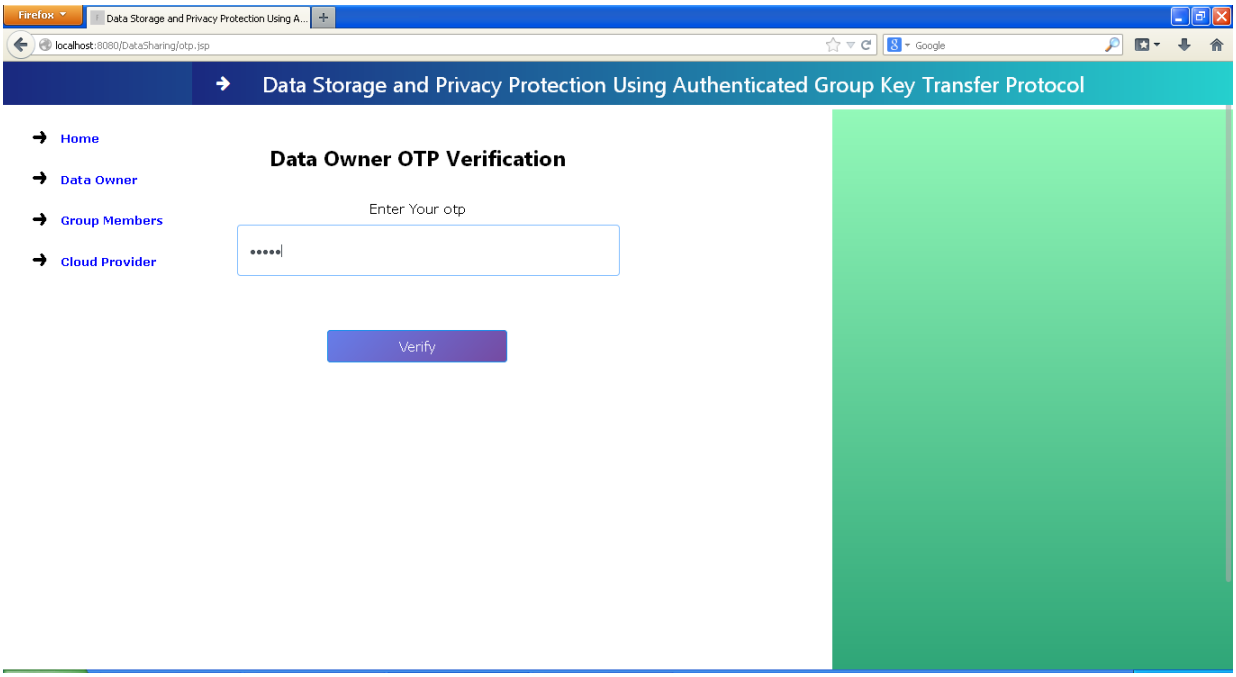

Fig 6.6 ENTER OTP

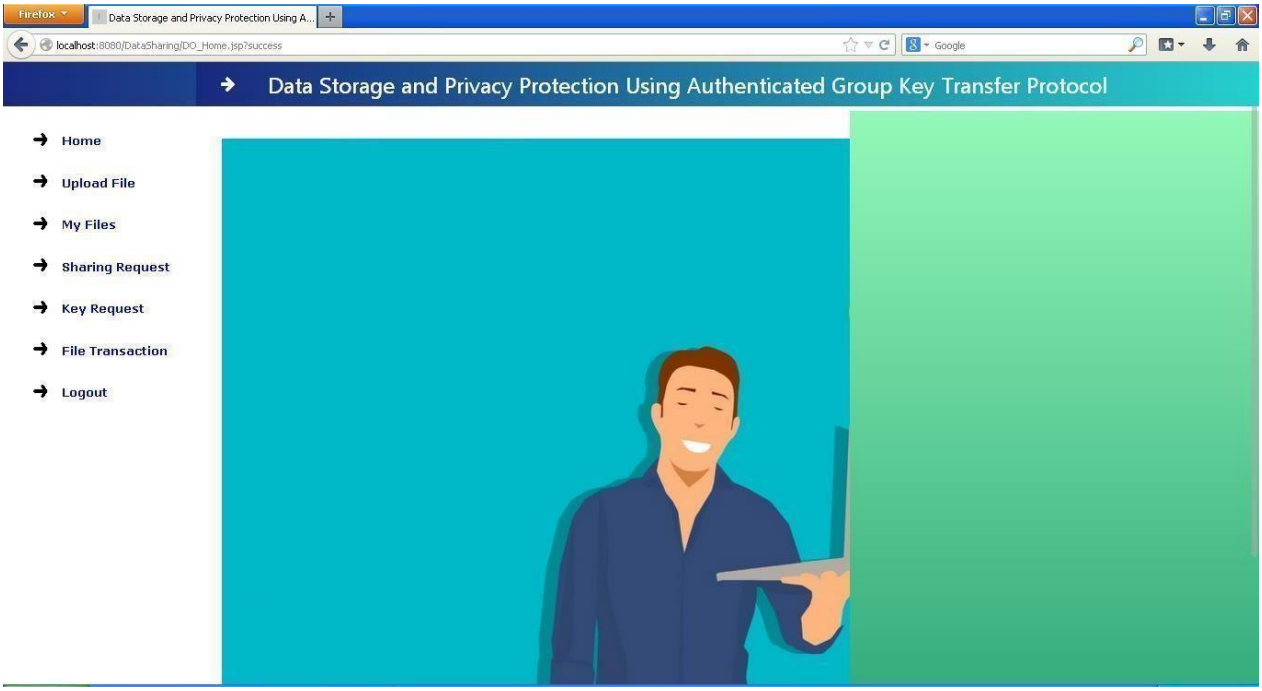

Fig 6.7 AFTER ENTER OTP

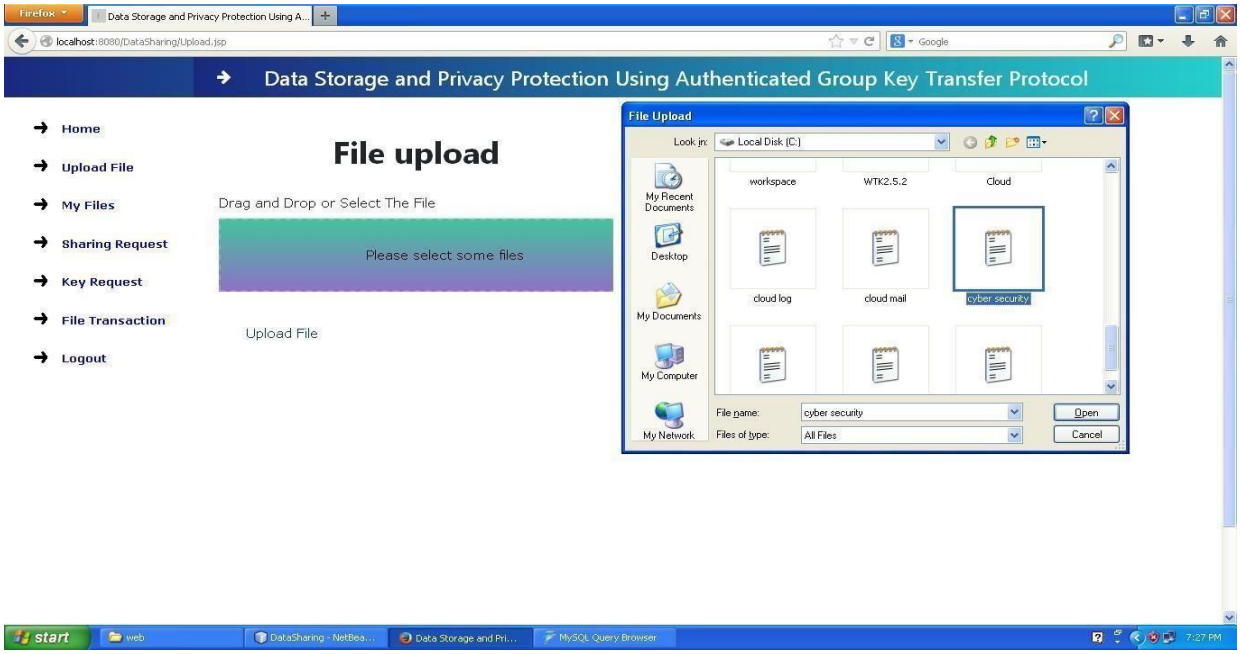

Fig 6.8 CHOOSE FILE TO UPLOAD

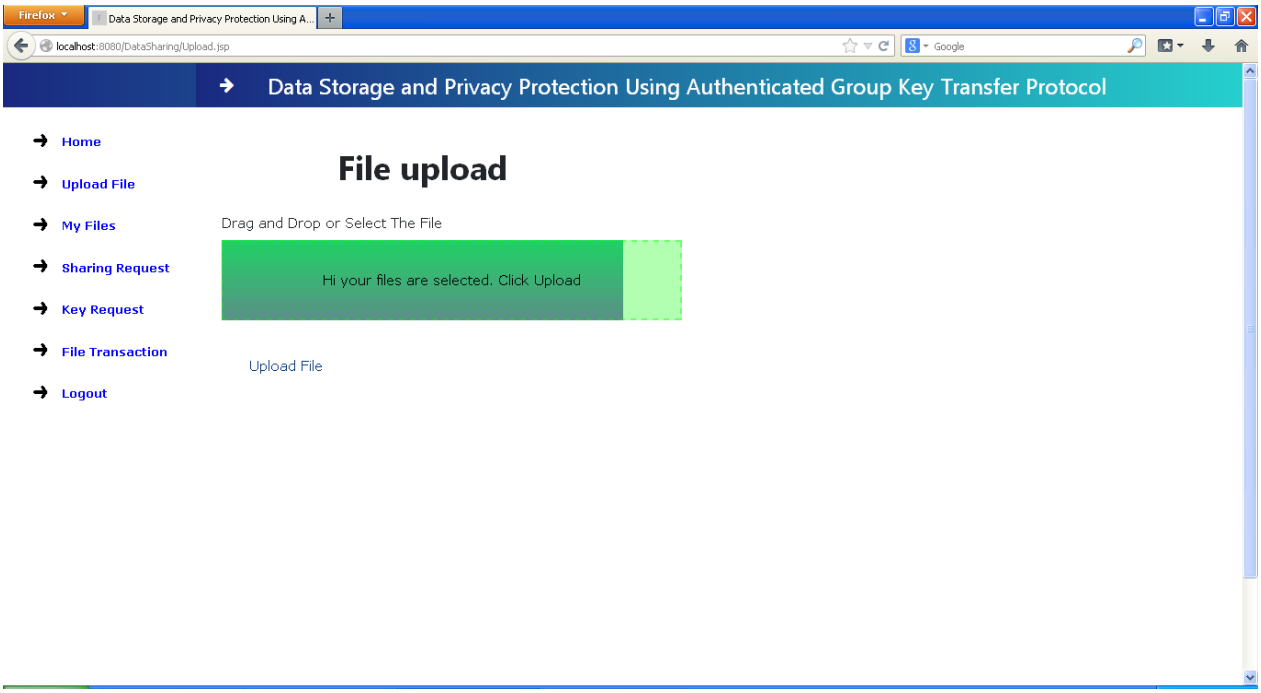

Fig 6.9 UPLOAD FILE

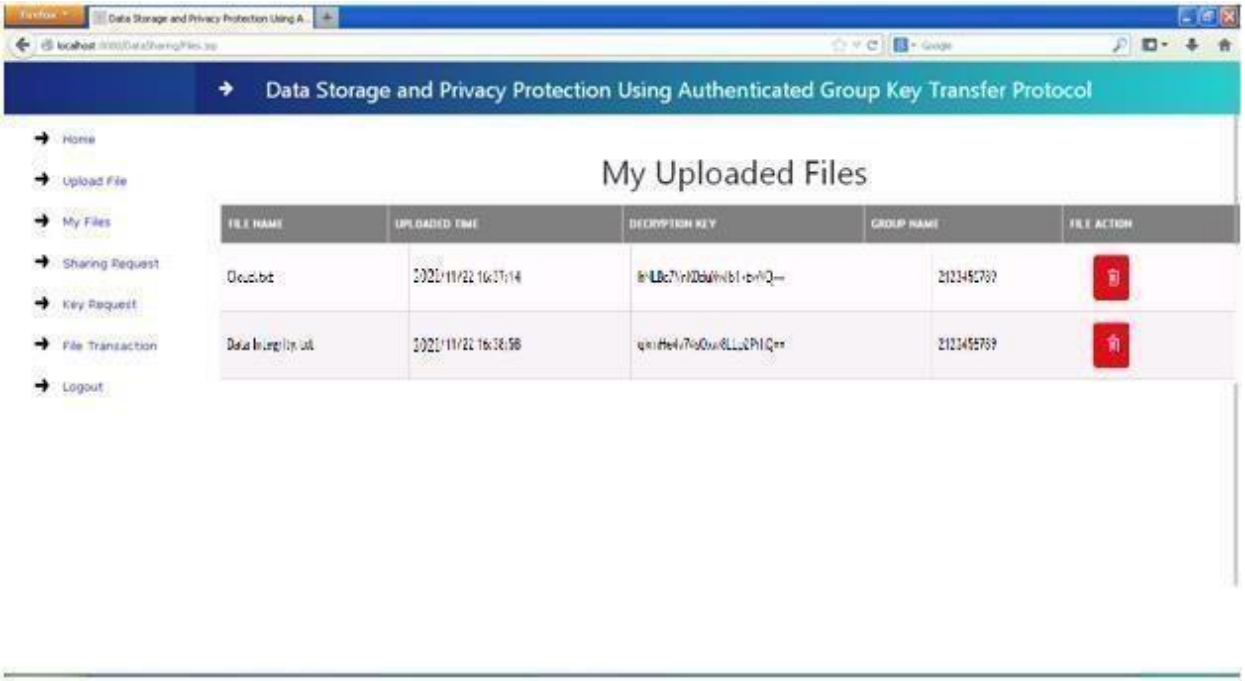

Fig 6.10 MY FILES

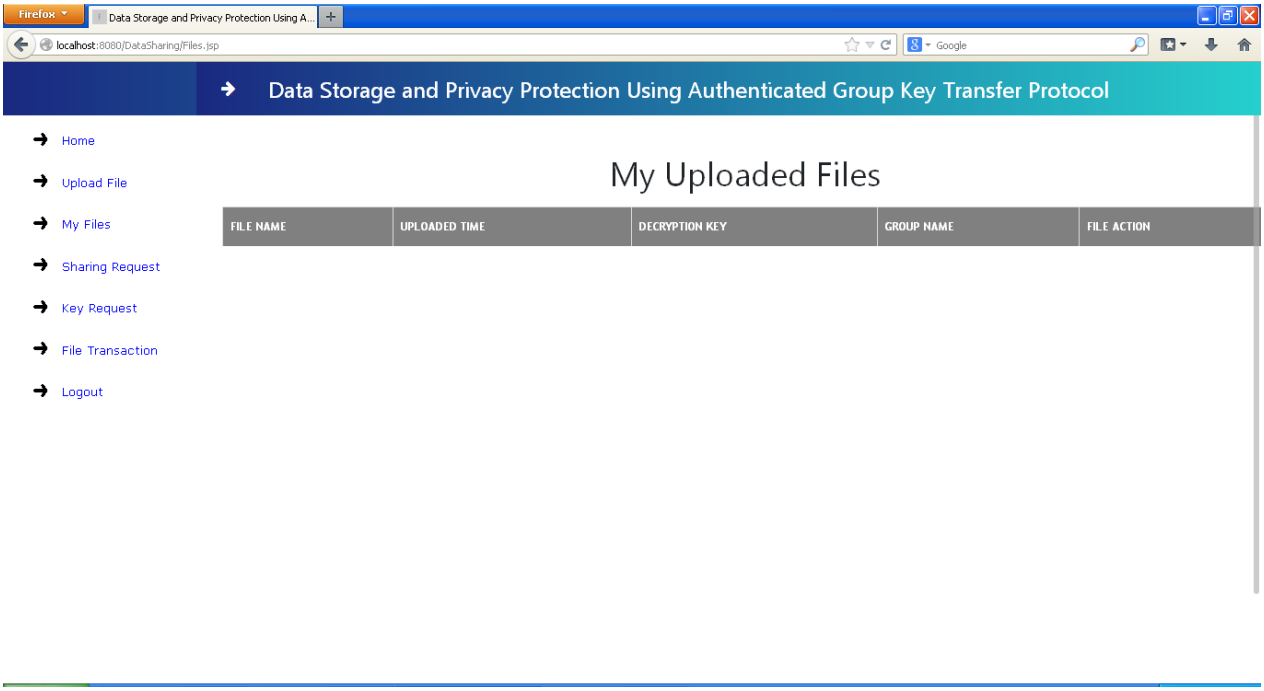

Fig 6.11 UPLOADED FILES

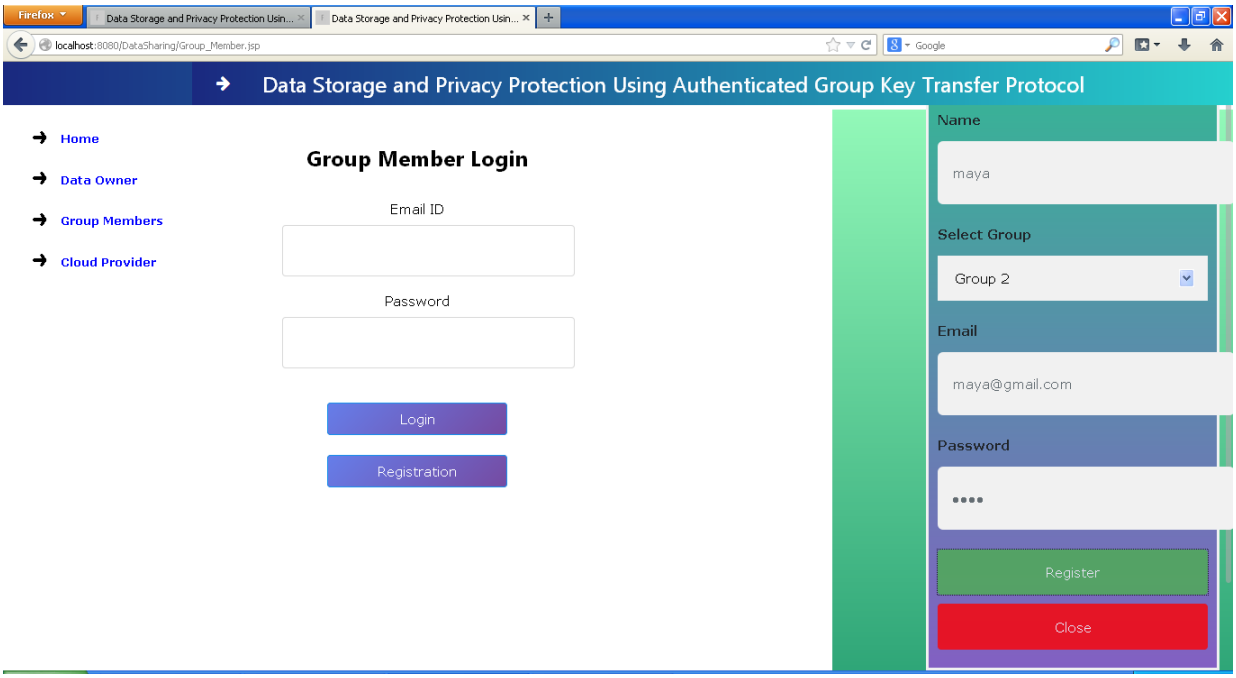

Fig 6.12 GROUP MEMBER REGISTER

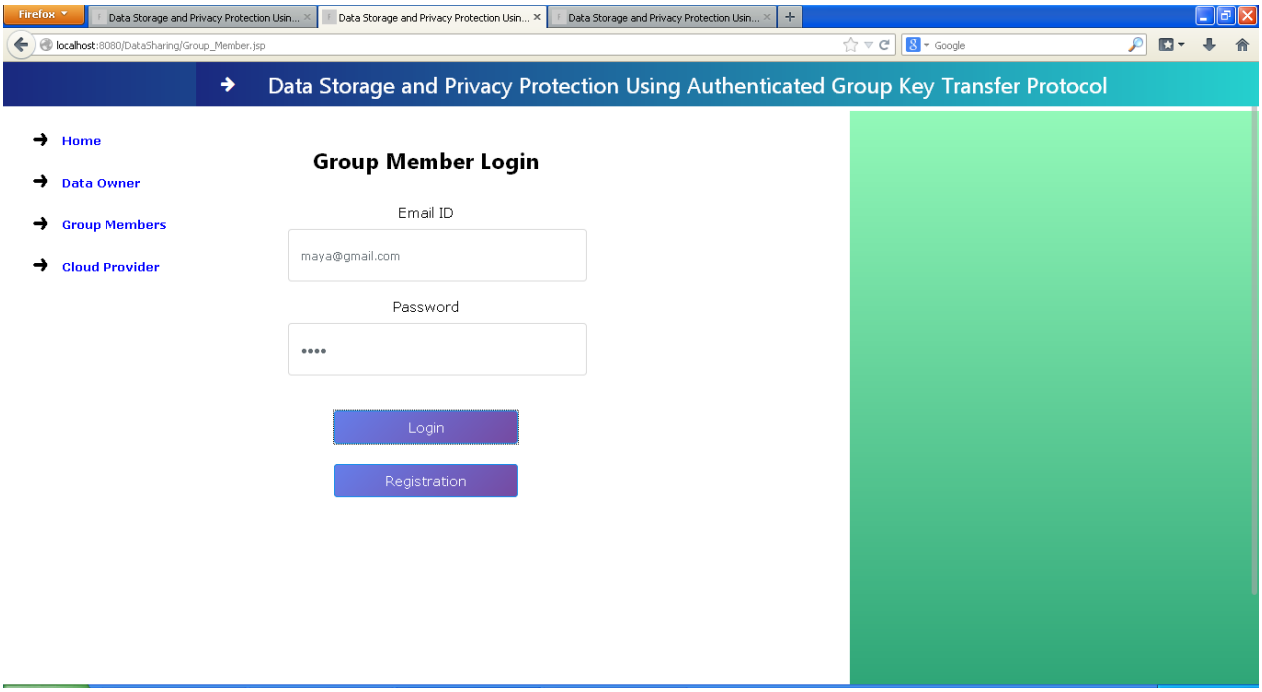

Fig 6.13 GROUP MEMBER LOGIN

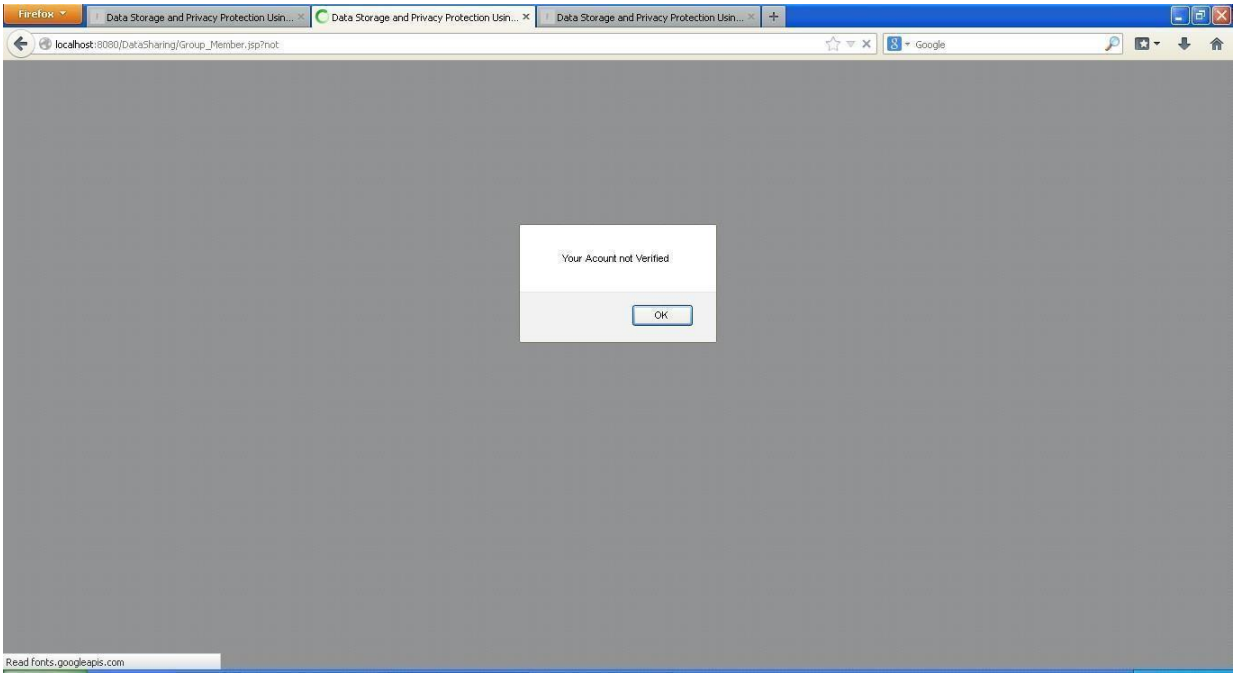

Fig 6.14 ACCOUNT NOT VERIFIED

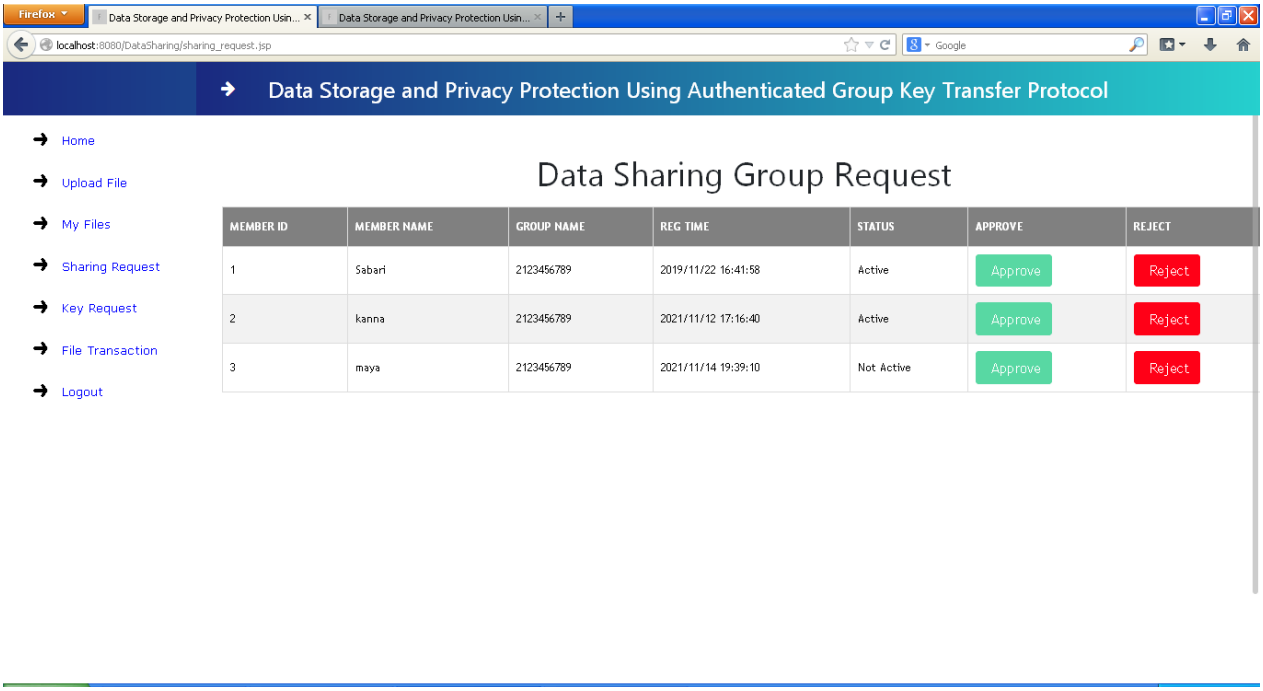

Fig 6.15 DATA SHARING

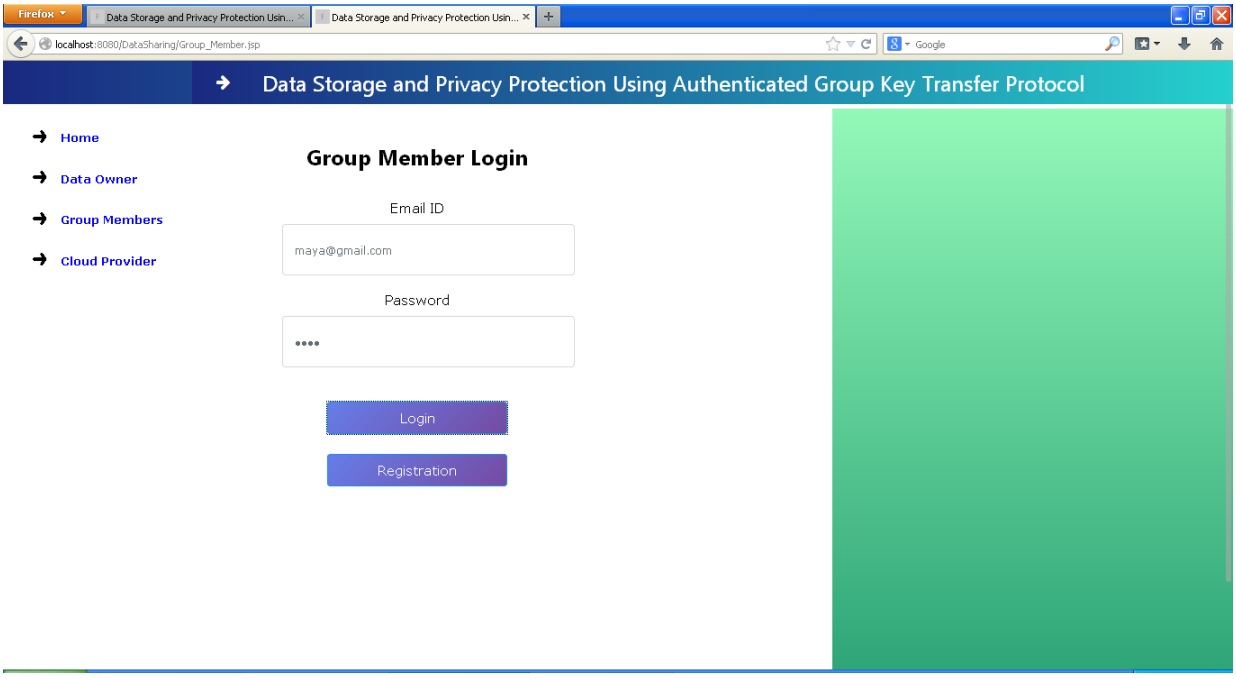

Fig 6.16 GROUP MEMBER LOGIN

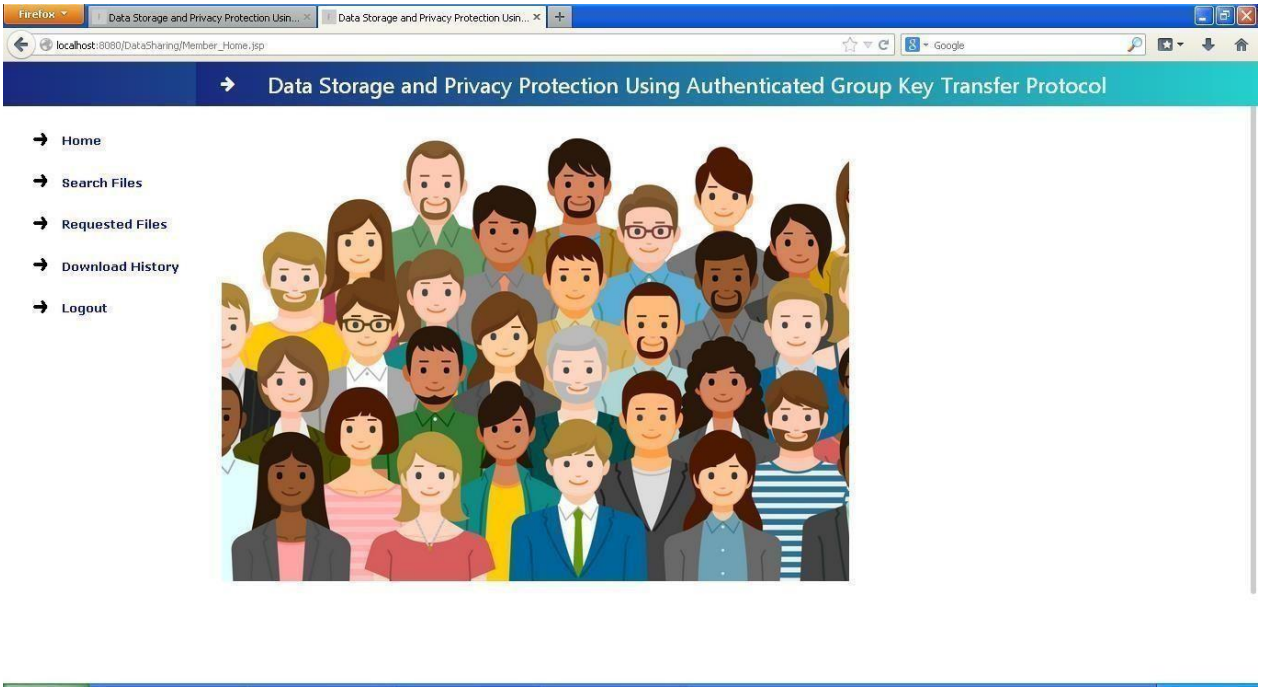

Fig 6.17 AFTER GROUP MEMBER LOGIN

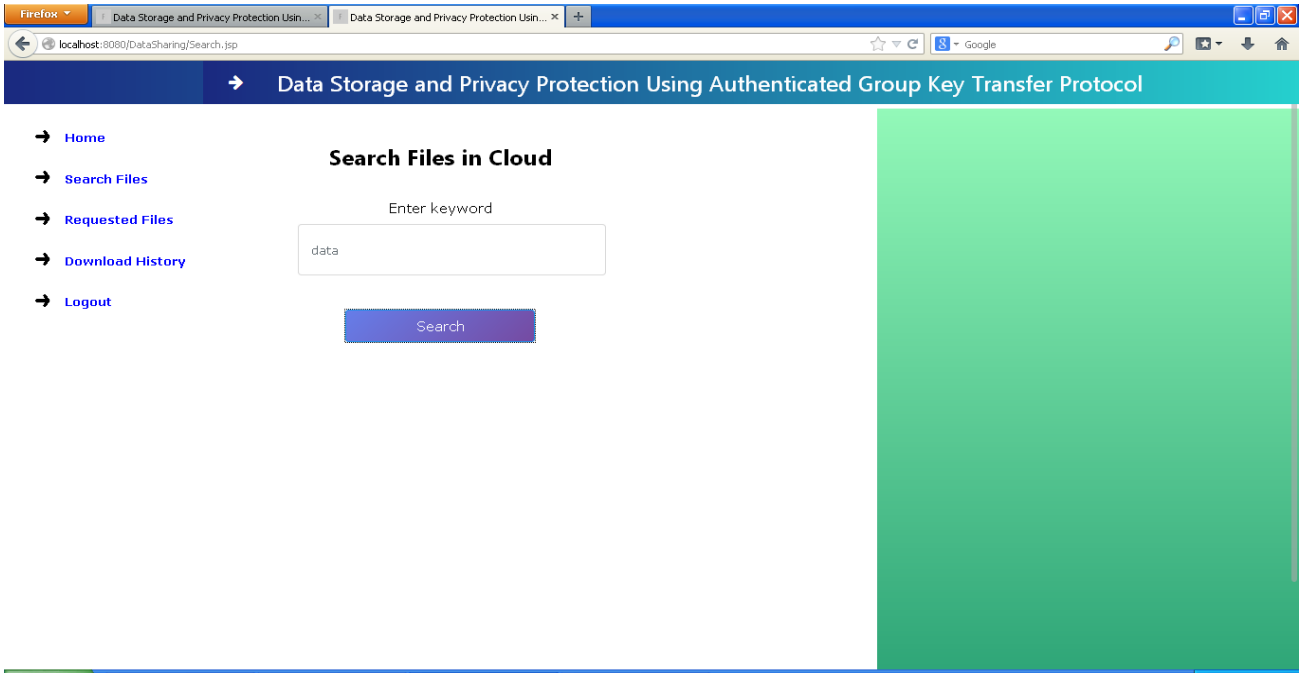

Fig 6.18 SEARCH FILES

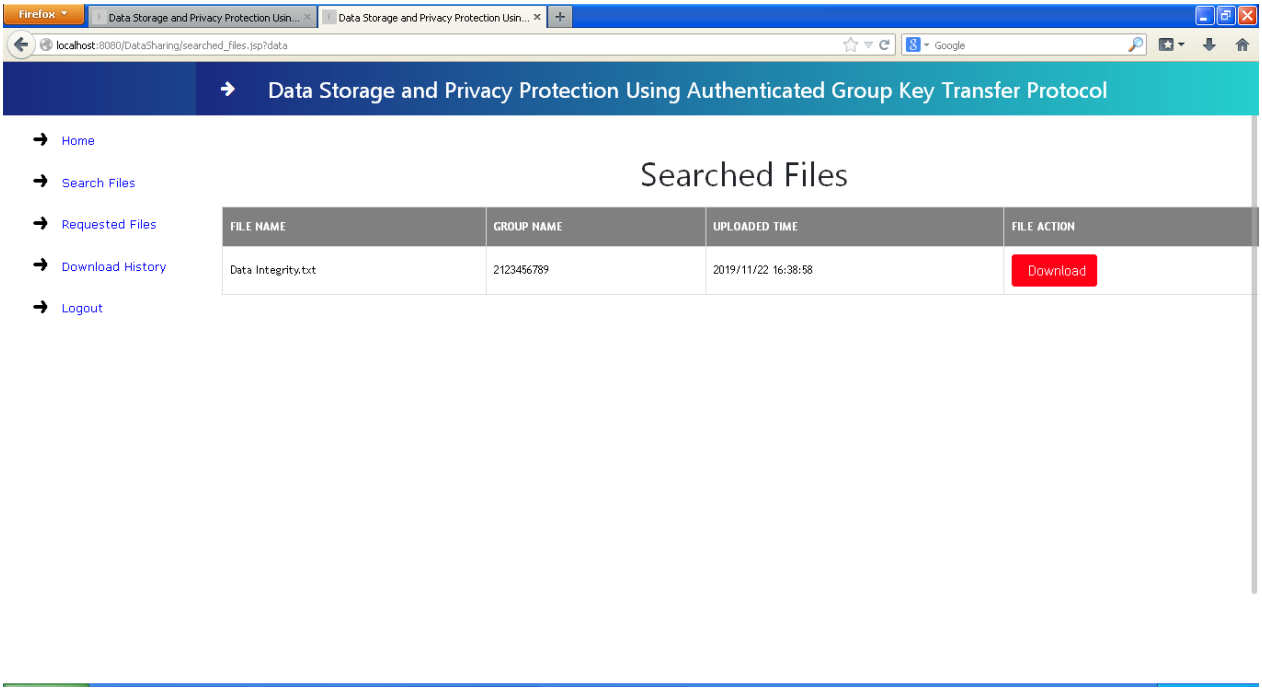

Fig 6.19 SEARCHED FILES

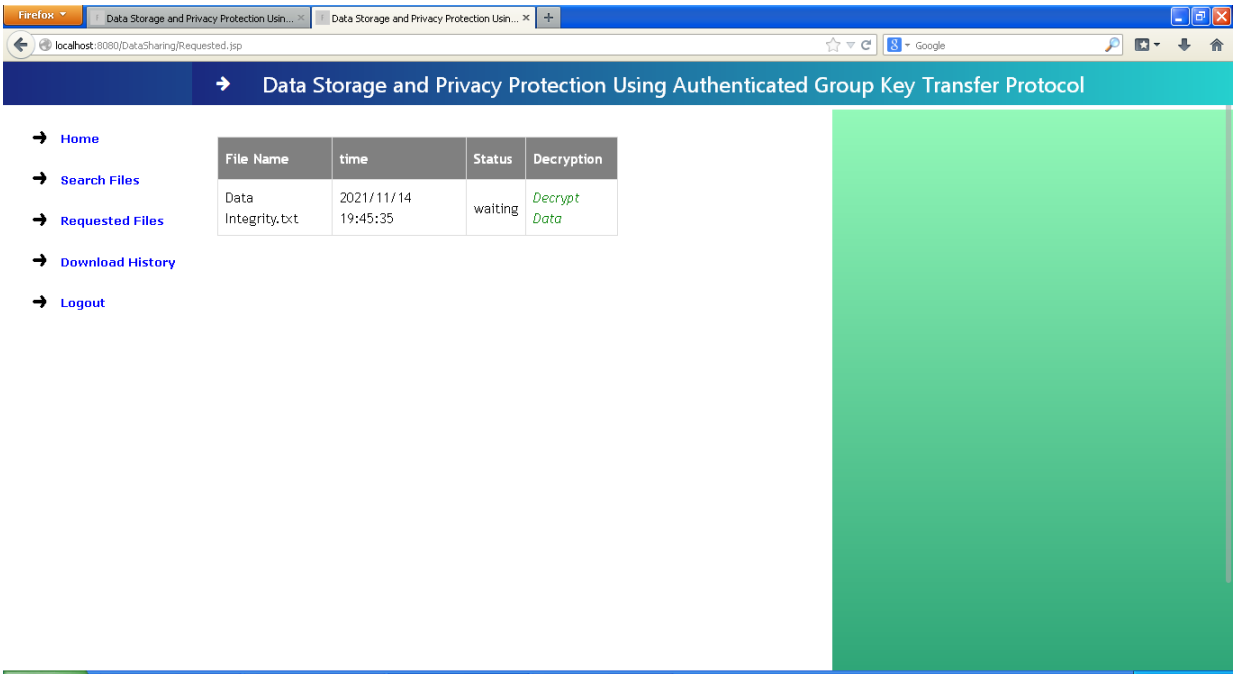

Fig 6.20 REQUESTED FILE

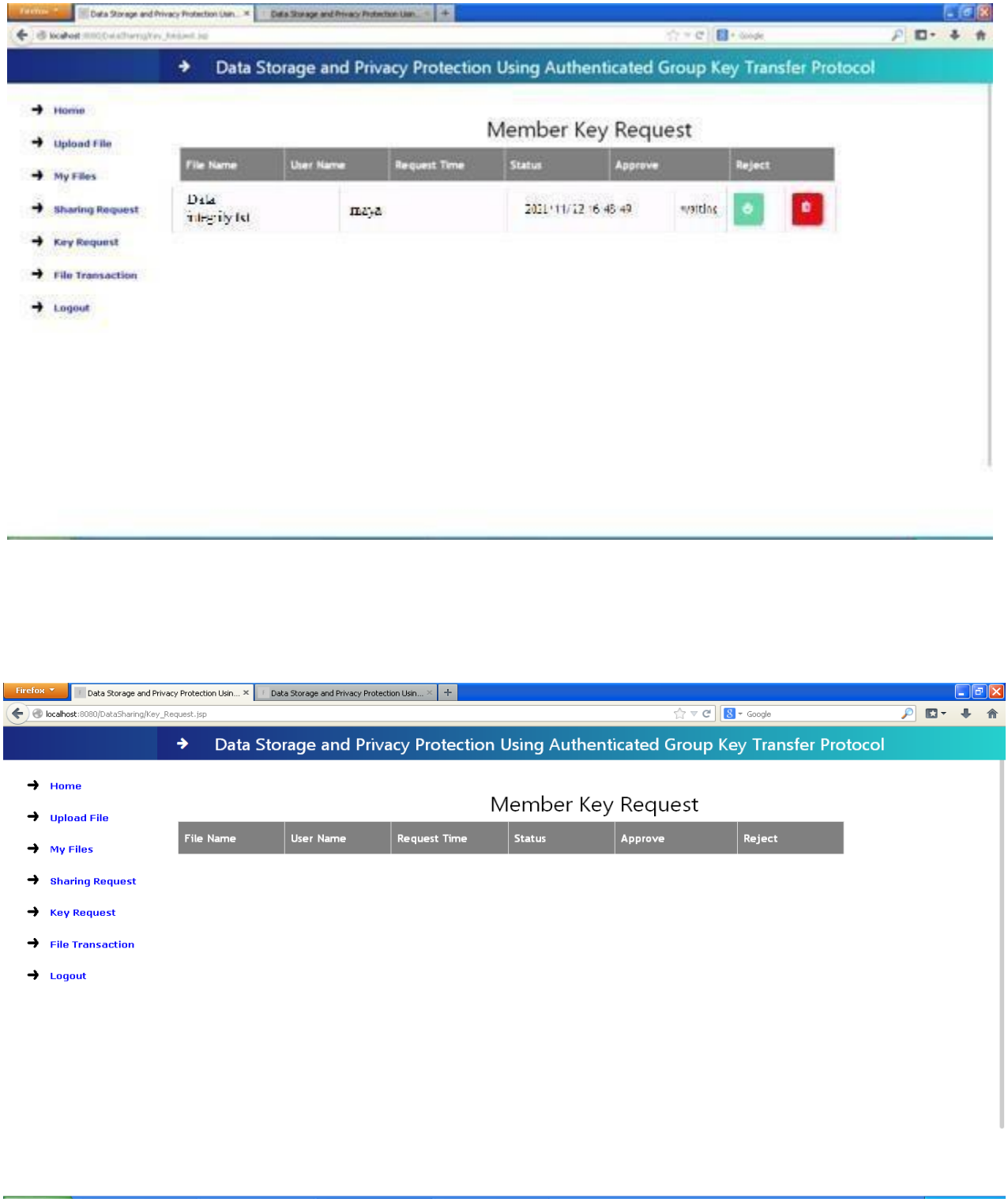

Fig 6.21 MEMBER KEY REQUEST

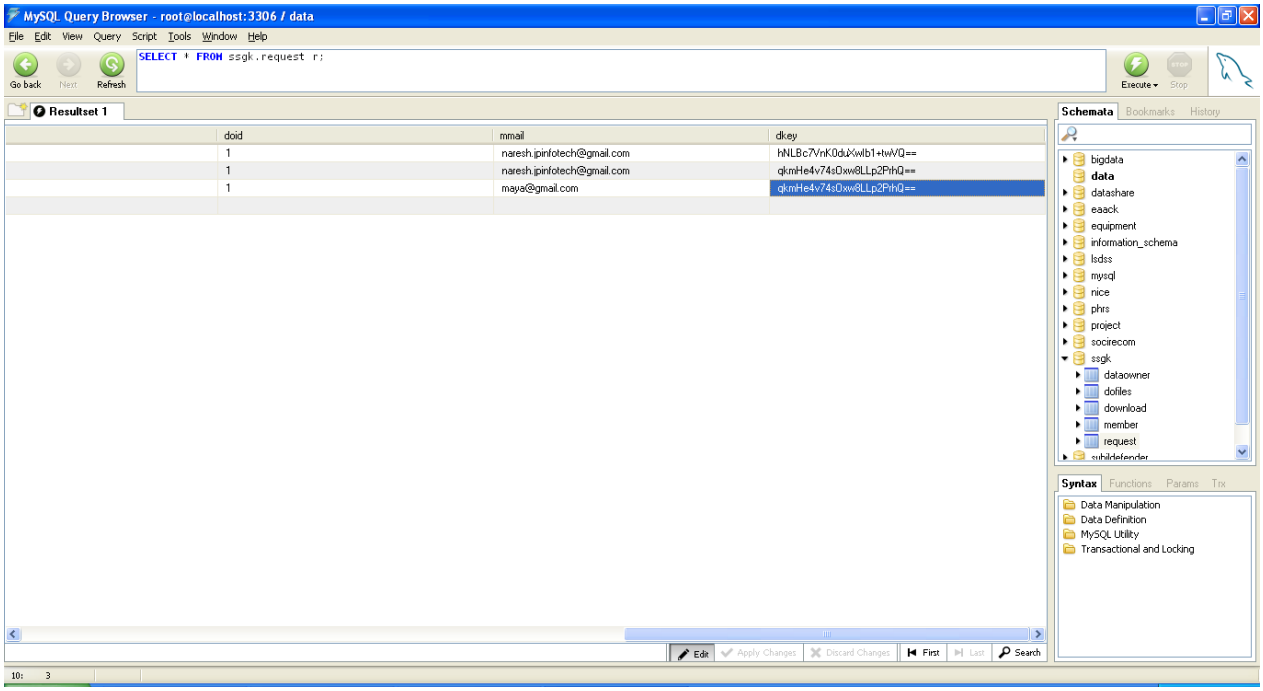

Fig 6.22 AFTER MEMBER KEY REQUEST

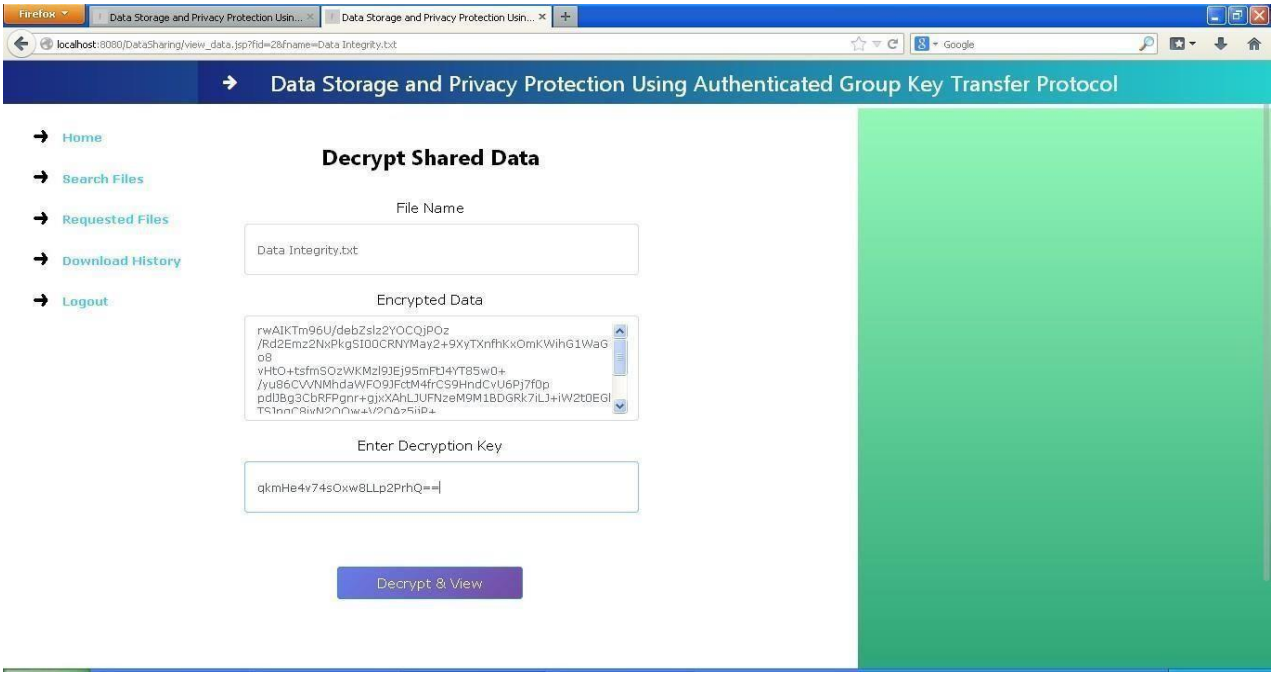

Fig 6.23 ENTER KEY TO DECRYPT & DOWNLOAD

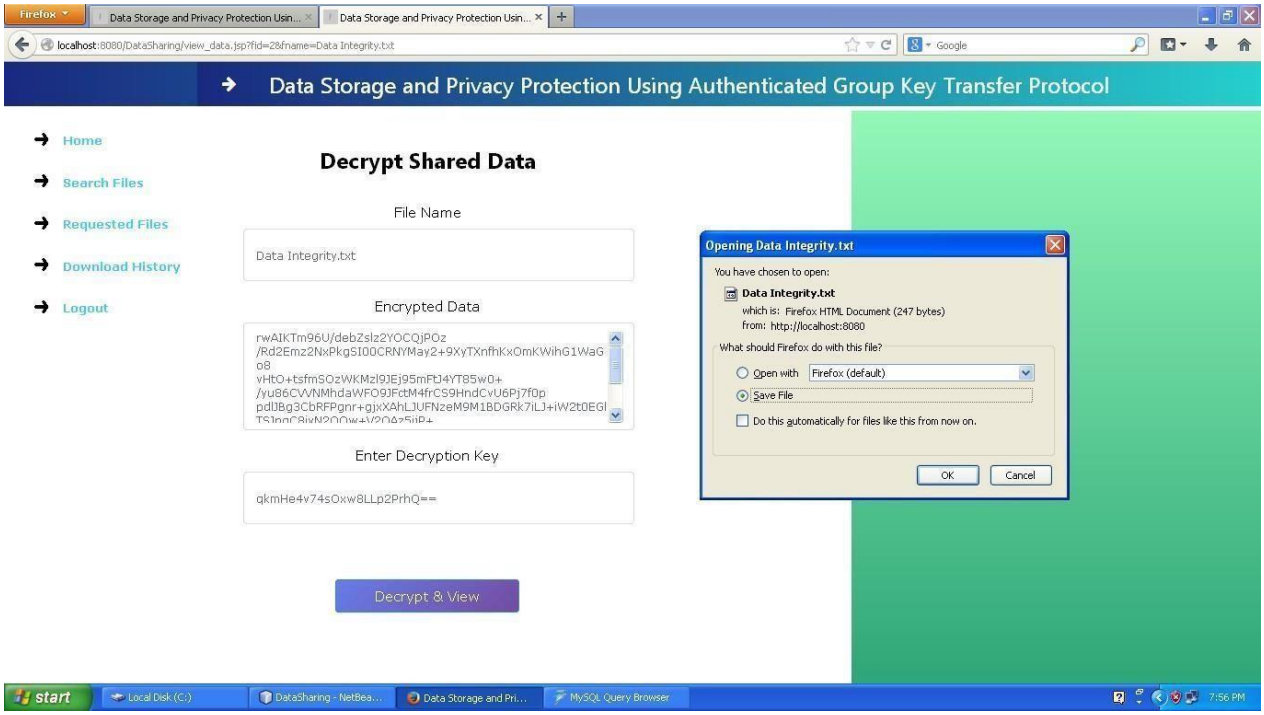

Fig 6.24 DECRYPT SHARED DATA

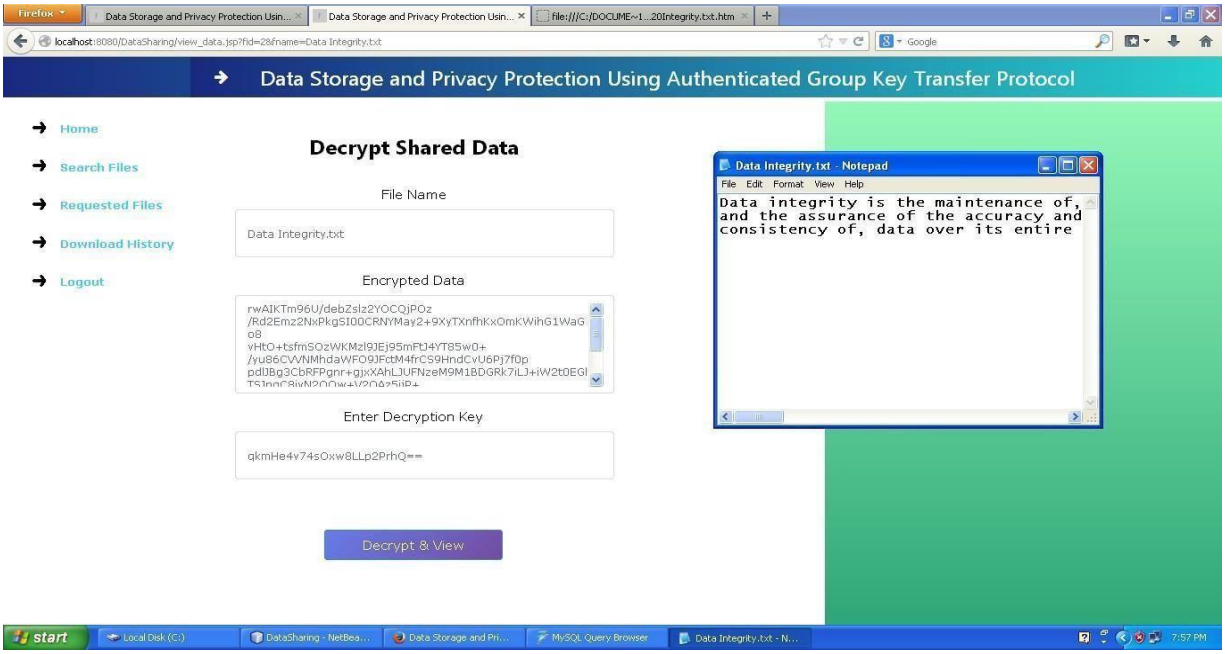

Fig 6.25 VIEW FILE

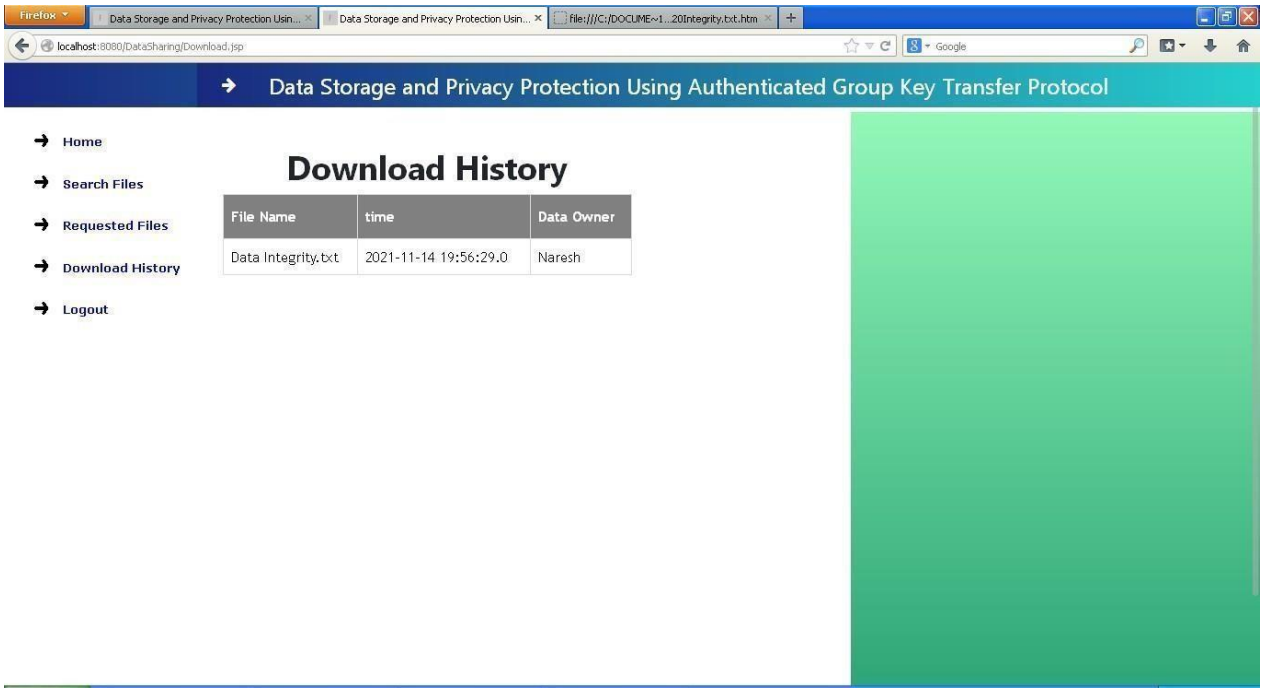

Fig 6.26 DOWNLOAD HISTORY

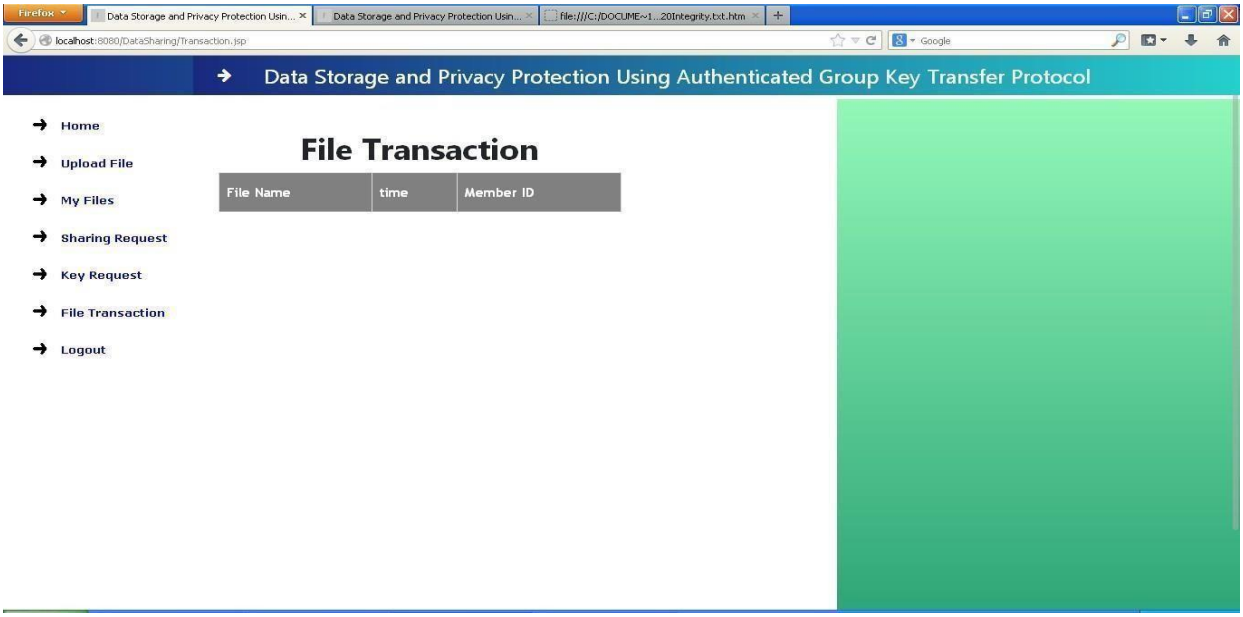

Fig 6.27 FILE TRANSACTION IN DATA OWNER

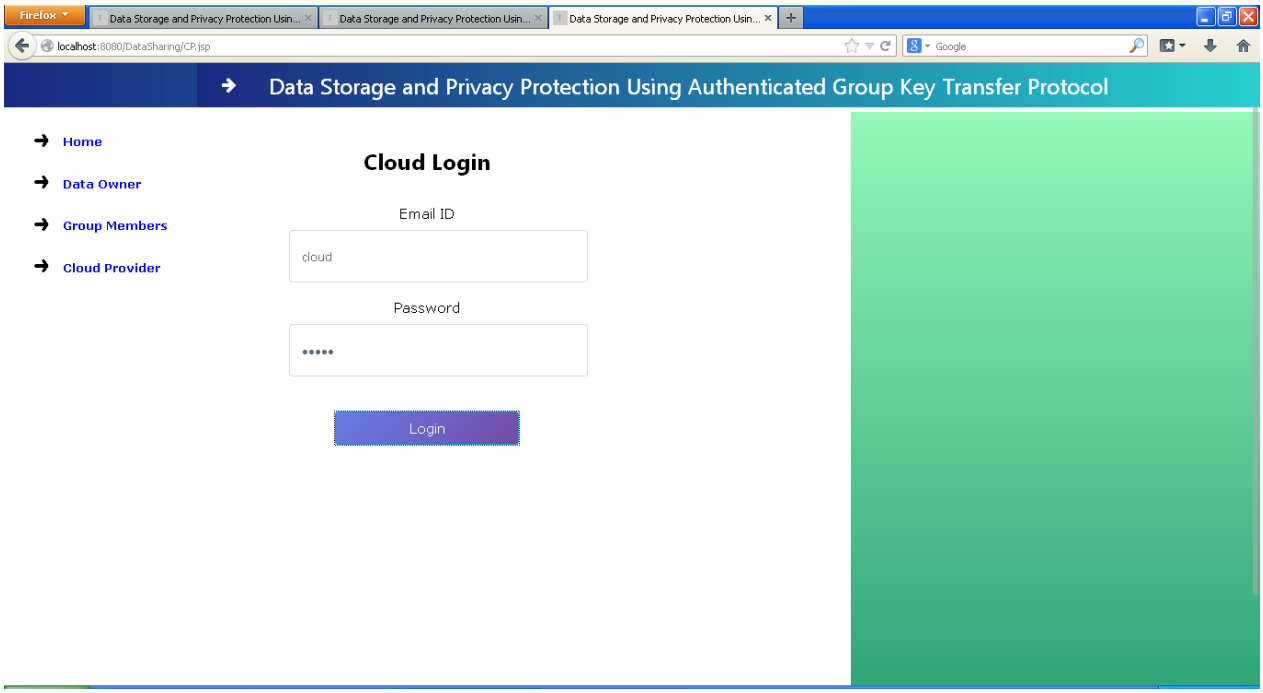

Fig 6.28 CLOUD LOGIN

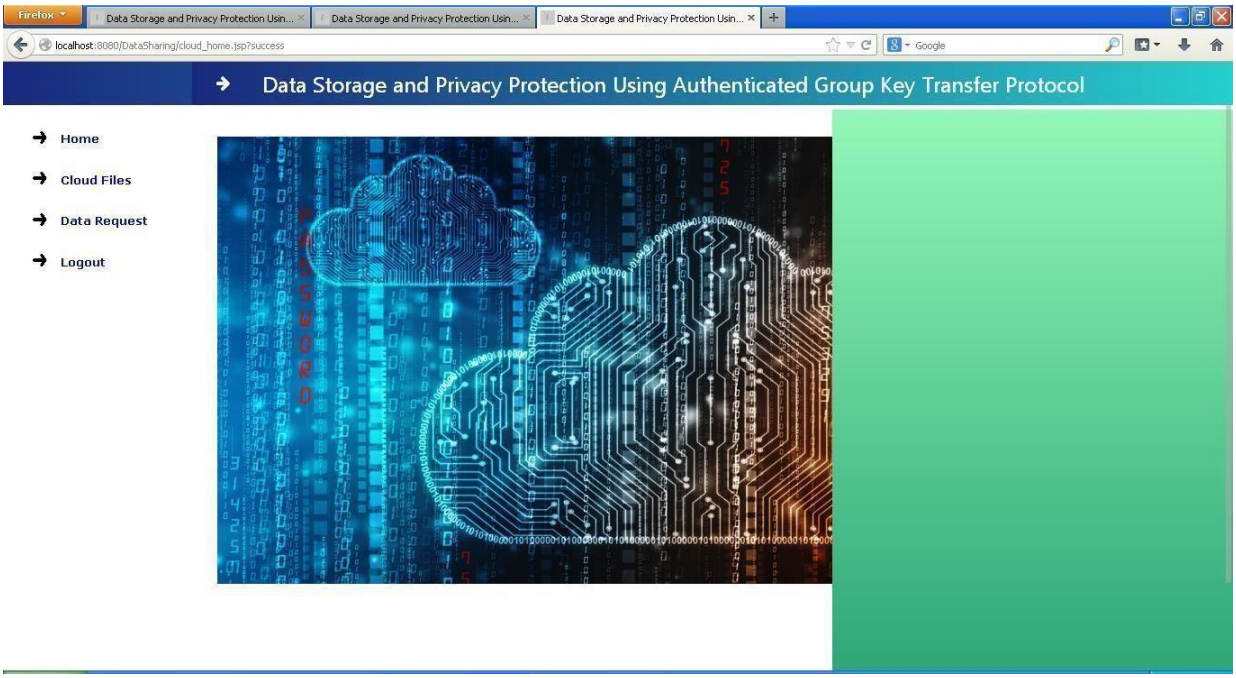

Fig 6.29 CLOUD HOME PAGE

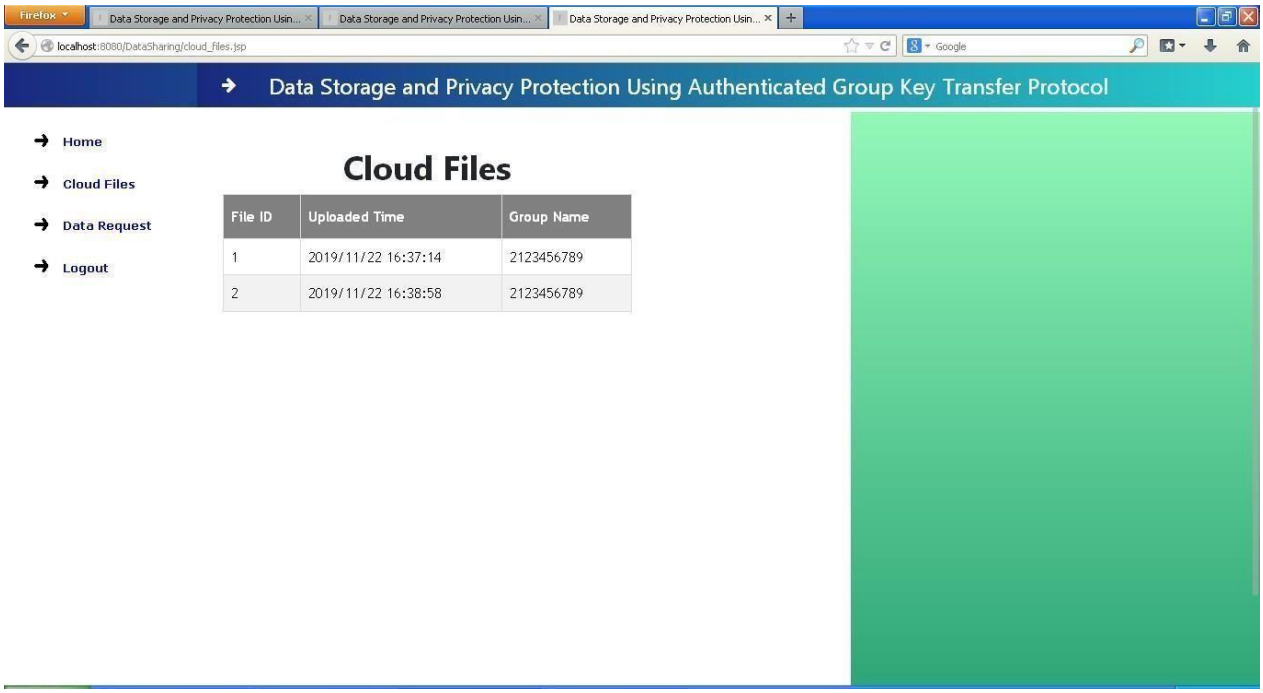

Fig 6.30 CLOUD FILES

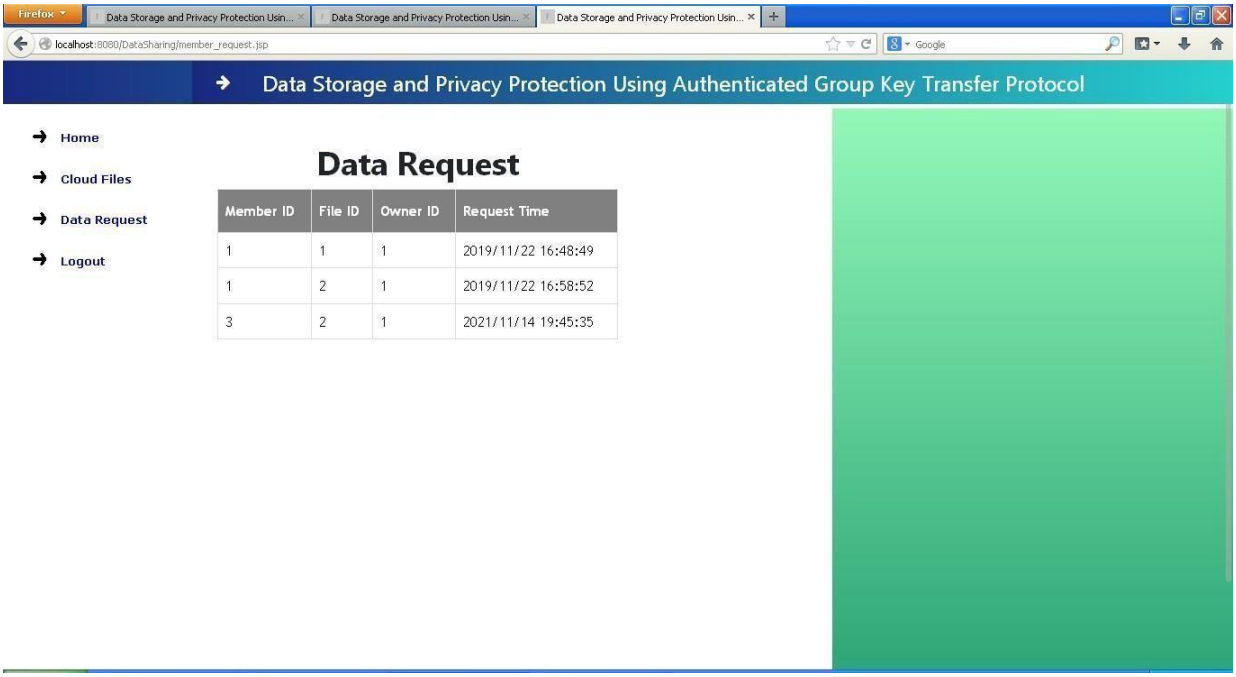

Fig 6.31 DATA REQUEST

6.1 USER DOCUMENTATION

Data owner.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Data Storage and Privacy Protection Using Authenticated Group Key Transfer Protocol</title>
    <link rel="stylesheet" href="node_modules/font-awesome/css/font-awesome.min.css" />
    <link rel="stylesheet" href="node_modules/perfect-scrollbar/dist/css/perfect-scrollbar.min.css" />
    <link rel="stylesheet" href="css/style.css" />
    <link rel="shortcut icon" href="images/favicon.png" />
  </head>
  <style>
    /* Button used to open the contact form - fixed at the bottom of the page */
    .open-button {
      background-color: #555;
      color: white;
      padding: 16px
      20px; border:
      none; cursor:
      pointer; opacity:
      0.8; position: fixed;
      bottom: 23px;
      right: 28px;
      width: 280px;
    }
    .form-popup {
      display: none;
      position:
      fixed; bottom:
      0; right: 15px;
      border: 3px solid
      #f1f1f1; z-index: 9;
    }
    .form-container {
      max-width:
      300px; padding:
      10px;
      background-color: white;
```

```
    }
    .form-container input[type=text], .form-container input[type=password] , .form-container select
      { width: 90%;
      padding: 15px;
      margin: 5px 0 22px 0;
      border: none;
      background: #f1f1f1;
    }
    .form-container input[type=text]:focus, .form-container input[type=password]:focus
      { background-color: #ddd;
      outline: none;
    }
    .form-container .btn {
      background-color:
      #4CAF50; color: white;
      padding: 16px
      20px; border:
      none; cursor:
      pointer; width:
      100%;
      height: 90%;
      margin-bottom:10px;
      opacity: 0.8;
    }
    .form-container .cancel
      { background-color:
      red;
    }
    .form-container .btn:hover, .open-button:hover
      { opacity: 1;
    }
    .blinking{
  animation:blinkingText 0.8s infinite;
}
@keyframes blinkingText{ }
}
  </style>
  <style>
article,aside,details,figcaption,figure,footer,header,main,menu,nav,section,summary{display:block}
  </style>
  <body>
```

```
<div class=" container-scroller">
  <!-- partial:partials/_navbar.html -->
  <nav class="navbar navbar-default col-lg-12 col-12 p-0 fixed-top d-flex flex-row " style="background: #1A2980;  /* fallback for old browsers */
background: -webkit-linear-gradient(to left, #26D0CE, #1A2980background: linear-gradient(to left, #26D0CE, #1A2980"
>
    <div class="bg-white text-center navbar-brand-wrapper">
      <!-- <a class="navbar-brand brand-logo" href="index.html"><img src="images/logo_star_black.png" /></a>
        <a class="navbar-brand brand-logo-mini" href="index.html"><img src="images/logo_star_mini.jpg" alt=""></a>-->
    </div>
    <div class="navbar-menu-wrapper d-flex align-items-center" style="background: #1A2980; background: -webkit-linear-gradient(to left, #26D0CE, #1A2980);
background: linear-gradient(to left, #26D0CE, #1A2980); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */
      <button class="navbar-toggler navbar-toggler d-none d-lg-block navbar-dark align-self-center mr-3" type="button" data-toggle="minimize">
        <span style="color: white" class="fa fa-arrow-right"></span>
      </button>
      <h3 class="center-aligned-table" style="color: white">Data Storage and Privacy Protection Using Authenticated Group Key Transfer Protocol</h3>
      <!-- <form class="form-inline mt-2 mt-md-0 d-none d-lg-block">
          <input class="form-control mr-sm-2 search" type="text" placeholder="Search">
        </form>-->
      <ul class="navbar-nav ml-lg-auto d-flex align-items-center flex-row">
        <!-- <li class="nav-item">
            <a class="nav-link profile-pic" href="#"><img class="rounded-circle" src="images/face.jpg" alt=""></a>
          </li>-->
        <!-- <li class="nav-item">
            <a class="nav-link" href="#"><i class="fa fa-th"></i></a>
          </li>-->
      </ul>
      <!-- <button class="navbar-toggler navbar-dark navbar-toggler-right d-lg-none align-self- center" type="button" data-toggle="offcanvas">
          <span class="navbar-toggler-icon"></span>
        </button>-->
    </div>
  </nav>
  <!-- partial -->
```

```
        <div class="container-fluid">
          <div class="row row-offcanvas row-offcanvas-right" style="background: white">
            <!-- partial:partials/_sidebar.html -->
            <nav class="sidebar sidebar-offcanvas" id="sidebar"style="background: white">
              <ul class="nav">
                <li class="nav-item active">
                  <a class="nav-link" href="index.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
                    <span class="menu-title"><b style="color: #0000FF ">Home</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Data_Owner.jsp">
            <img src="images/arrow.png" width="200px" alt="" style="color: #33D4FF ">
<span class="menu-title"><b style="color: #0000FF ">Data Owner</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Group_Member.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
    <span class="menu-title"><b style="color: #0000FF ">Group Members</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="CP.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
    <span class="menu-title"><b style="color: #0000FF ">Cloud Provider</b></span>
                  </a></li>
            <div class="content-wrapper" style="background: white">
              <%
       if (request.getParameter("success") != null) {
     %>
     <script>alert('User Registration successful');</script>
     <%          }%>
              <center><h3 style="color: black"><b>Data Owner Login</b></h3>
                <form action="DO_login.jsp" method="get" >
                  <label style="color: black">Email ID</label>
                  <div class="form-group" style="width: 300px" >
 <input type="text" name="email" required="" class="form-control p_input"auto="off" >
                  </div>
                  <label style="color: black">Password</label>
                  <div class="form-group" style="width: 300px" >
```

```
                    <input type="password" name="pass" required="" class="form-control p_input" >
                  </div>
                  <div class="text-center" style="width: 200px" >
                    <button type="submit" class="btn btn-primary btn-block enter-
btn" style="background-image: linear-gradient(135deg, #667eea 0%, #764ba2
100%);"><a>Login</a></button>
                  </form>
                </center>
                <div>
                </div>
                <div class="form-popup" id="myForm">
                  <form action="DO_signup.jsp" method="get"  class="form-container">
                    <label for="email"><b>Name</b></label>
                    <input type="text"  placeholder="Enter Name"  name="name" autocomplete="off"
required>
                    <label for="email"><b>Email</b></label>
                    <input type="text"  placeholder="Enter Email"  name="email" autocomplete="off"
required>
                    <label for="psw"><b>Select Group</b></label>
                    <select name="group" required>
                      <option value="112456789">Group 1</option>
                      <option value="2123456789">Group 2</option>
                      <option value="3123456789">Group 3</option>
                    </select>
                    <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="password" required>
                    <button type="submit" class="btn">Register</button>
                    <button type="button" class="btn cancel" onclick="closeForm()">Close</button>
                  </form>
                </div></center>
              </div>
  </body>
</html>
```

Data owner signup.jsp

```
<%@page import="java.security.SecureRandom"%>
<%@page import="java.util.Random"%>
<%@page import="javax.el.ELException"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page import="java.util.Date"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
```

```
<%
   String name = request.getParameter("name");
   String mail = request.getParameter("email");
   String pass = request.getParameter("password");
   String rpass = request.getParameter("password");
   String grp = request.getParameter("group");

   TDES tdes = new TDES();
   String passEncryption = tdes.encrypt(pass);
   TDES tdes1 = new TDES();
   String passEncryption1 = tdes1.encrypt(rpass);

   System.out.println("passEncryption---------->>        :" + passEncryption);
   System.out.println("\npassEncryption1---------->>           :" +
   passEncryption1); DateFormat dateFormat = new
   SimpleDateFormat("YYYY/MM/dd HH:mm:ss"); Date date = new Date();
   String time = dateFormat.format(date);
   System.out.println("Date and Time : " + time);
   Statement st = conn.createStatement();
   try {
      int i = st.executeUpdate("insert into dataowner( Name, Email, Password, RPassword, Time,OTP,
grp)values('" + name + "','" + mail + "','" + passEncryption + "','" + passEncryption1 + "','" + time +
"','Waiting','" + grp + "') ");
      if       (i       !=       0)       {
         System.out.println("success");
         response.sendRedirect("Data_Owner.jsp?success");
      } else {
         System.out.println("Data_Owner.jsp?failed");
      }
   } catch (Exception ex)
      {
      ex.printStackTrace();
   }
%>
```

Key requet.jsp

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="SSGK.SQLconnection"%>
<%@page contentType="text/html"  pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Data Storage and Privacy Protection Using Authenticated Group Key Transfer Protocol</title>
  <link rel="stylesheet" href="node_modules/font-awesome/css/font-awesome.min.css" />
  <link rel="stylesheet" href="node_modules/perfect-scrollbar/dist/css/perfect-scrollbar.min.css" />
  <link rel="stylesheet" href="css/style.css" />
  <link rel="shortcut icon" href="images/favicon.png" />
</head>

<style>
<body>
  <div class=" container-scroller">
    <!-- partial:partials/_navbar.html -->
    <nav class="navbar navbar-default col-lg-12 col-12 p-0 fixed-top d-flex flex-row " style="background: #1A2980; /* fallback for old browsers */
      <div class="bg-white text-center navbar-brand-wrapper">
        <!-- <a class="navbar-brand brand-logo" href="index.html"><img src="images/logo_star_black.png" /></a>
          <a class="navbar-brand brand-logo-mini" href="index.html"><img src="images/logo_star_mini.jpg" alt=""></a>-->
      </div>
      <div class="navbar-menu-wrapper d-flex align-items-center" style="background: #1A2980; background: linear-gradient(to left, #26D0CE, #1A2980); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */">
        <button class="navbar-toggler navbar-toggler d-none d-lg-block navbar-dark align-self-center mr-3" type="button" data-toggle="minimize">
          <span style="color: white" class="fa fa-arrow-right"></span>
        </button>
          </form>-->
        <ul class="navbar-nav ml-lg-auto d-flex align-items-center flex-row">
          <!-- <li class="nav-item">
            <a class="nav-link profile-pic" href="#"><img class="rounded-circle" src="images/face.jpg" alt=""></a>
          </li>-->
          <!-- <li class="nav-item">
            <a class="nav-link" href="#"><i class="fa fa-th"></i></a>
          </li>-->
        </ul>
```

```
            <!--          <button class="navbar-toggler navbar-dark navbar-toggler-right d-lg-none align-
self- center" type="button" data-toggle="offcanvas">
                    <span class="navbar-toggler-icon"></span>
                  </button>-->
          </div>
        </nav>

        <!-- partial -->
        <div class="container-fluid">
          <div class="row row-offcanvas row-offcanvas-right" style=" background: white">
            <!-- partial:partials/_sidebar.html -->
            <nav class="sidebar sidebar-offcanvas" id="sidebar"style="background: white">
              <ul class="nav">
                <li class="nav-item active">
                  <a class="nav-link" href="DO_Home.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
                    <span class="menu-title"><b style="color: #0000FF">Home</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Upload.jsp">
          <img src="images/arrow.png" width="200px" alt="" style="color: #33D4FF">
           <span class="menu-title"><b style="color: #0000FF">Upload File</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Files.jsp">
          <img src="images/arrow.png" width="200px" alt="" style="color: #33D4FF">
        <span class="menu-title"><b style="color: #0000FF">My Files</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="sharing_request.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
        <span class="menu-title"><b style="color: #0000FF">Sharing Request</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Key_Request.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
      <span class="menu-title"><b style="color: #0000FF">Key Request</b></span>
                  </a>
```

```
            </li>
            <li class="nav-item">
               <a class="nav-link" href="Transaction.jsp">
                  <img src="images/arrow.png" width="200px" alt="">
<span class="menu-title"><b style="color: #0000FF">File Transaction</b></span>
               </a>
            </li>
            <li class="nav-item">
               <a class="nav-link" href="index.jsp">
                  <img src="images/arrow.png" width="200px" alt="">
<span class="menu-title"><b style="color: #0000FF">Logout</b></span>
         </nav>
   <!-- partial -->
         <div class="content-wrapper" style="background: white">
            <h2 class="w3-center" style="color: black">Member Key Request</h2>
               <table id="customers">
                        <thead>
                           <tr>
                              <th>File Name</th>
                              <th>User Name</th>
                              <th>Request Time</th>
                               <th>Status</th>
                              <th>Approve</th>
                              <th>Reject</th>
                           </tr>
                        </thead>
                        <%
                           String oid = (String)session.getAttribute("doid");
                           Connection con = SQLconnection.getconnection();
                           Statement st = con.createStatement();
                           try {
    ResultSet rs = st.executeQuery("Select * from request where doid ='" +oid+ "'");
                              while (rs.next()) {
                        %>
                        <tbody>
                           <tr class="odd gradeX">
                        <form action="key_approve.jsp"  method="get">
                           <input hidden="" name="fid" value="<%=rs.getString("fid")%>"
                           type="text"

                           <input       hidden=""     name="mail"
                           value="<%=rs.getString("mmail")%>"
```

```
                                <input hidden="" name="time" value="<%=rs.getString("time")%>"
type="text" >
                                <td class="center"><input style="background-color: transparent; border:
                                none;
color: #000" name="filename" value="<%=rs.getString("filename")%>" type="text" ></td>
                                  <td class="center"><input style="background-color: transparent; border:
none; color: #000" name="mname" value="<%=rs.getString("mname")%>" type="text" ></td>
                                  <td class="center"><input style="background-color: transparent; border:
none; color: #000" value="<%=rs.getString("time")%>" type="text" ></td>
                                  <td class="center"><%=rs.getString("statuss")%> </td>
                                  <td><i><button type="submit" class="btn btn-success btn-circle fa
fa- check-circle fa-2x" style="color: white ; width: 40px; height: 40px"></button></i></td>
                              </form>
                              <td><i><a
href="key_reject.jsp?mid=<%=rs.getString("mid")%>&email=<%=rs.getString("mmail")%>&time=<%=rs.
getString("time")%>" class="btn btn-danger btn-circle fa fa-trash fa-2x " style="color: white; width:
40px; height: 40px"></a></i></td>
                                </tr>
                              </tbody>
                              <%                          }
                                } catch (Exception ex)
                                  {
                                  ex.printStackTrace()
                                  ;
          <!-- partial:partials/_footer.html -->
          <!-- partial -->
        </div>
      </div>
    </div>

    <script src="node_modules/jquery/dist/jquery.min.js"></script>
    <script src="node_modules/popper.js/dist/umd/popper.min.js"></script>
 scrollbar.jquery.min.js"></script>
    <script src="js/off-canvas.js"></script>
    <script src="js/hoverable-collapse.js"></script>
    <script src="js/misc.js"></script>
  </body>
</html>
```

Data owner –login.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.security.SecureRandom"%>
<%@page import="java.util.Random"%>
```

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

        <!-- partial -->
        <div class="container-fluid">
          <div class="row row-offcanvas row-offcanvas-right" style=" background: white">
            <!-- partial:partials/_sidebar.html -->
            <nav class="sidebar sidebar-offcanvas" id="sidebar"style="background: white">
              <ul class="nav">
                <li class="nav-item active">
                  <a class="nav-link" href="DO_Home.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
                    <span class="menu-title"><b style="color: #0000FF">Home</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Upload.jsp">
            <img src="images/arrow.png" width="200px" alt="" style="color: #33D4FF">
             <span class="menu-title"><b style="color: #0000FF">Upload File</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Files.jsp">
            <img src="images/arrow.png" width="200px" alt="" style="color: #33D4FF">
          <span class="menu-title"><b style="color: #0000FF">My Files</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="sharing_request.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
          <span class="menu-title"><b style="color: #0000FF">Sharing Request</b></span>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Key_Request.jsp">

                <li class="nav-item">
                  <a class="nav-link" href="Transaction.jsp">
                    <img src="images/arrow.png" width="200px" alt="">
        <span class="menu-title"><b style="color: #0000FF">File Transaction</b></span>
                  </a>
                </li>
```

```
<%@page import="java.sql.Connection"%>
<%
   String mail =
   request.getParameter("email"); String pass
   = request.getParameter("pass"); TDES
   tdes = new TDES();
   String passEncryption = tdes.encrypt(pass);

   System.out.println("Check User name And Password : " + mail + pass);
   Connection con = SQLconnection.getconnection();
   Statement st =
   con.createStatement(); Statement sto
   = con.createStatement(); try {
      ResultSet rs = st.executeQuery("SELECT * FROM dataowner where email='" + mail + "' AND
password='" + passEncryption + "' ");
      if (rs.next()) {
         session.setAttribute("doname", rs.getString("Name"));
         session.setAttribute("domail", rs.getString("Email"));
         session.setAttribute("doid", rs.getString("id"));
         session.setAttribute("grp", rs.getString("grp"));

         Random RANDOM = new SecureRandom();
      int PASSWORD_LENGTH = 5;
      String letters = "123456789";
      String otp = "";
      for (int i = 0; i < PASSWORD_LENGTH; i++) {
         int index = (int) (RANDOM.nextDouble() *
         letters.length()); otp += letters.substring(index, index + 1);
      }
      String DO_OTP = "" + otp;
       int i = sto.executeUpdate("update dataowner set OTP ='"+DO_OTP+"' where email='" +mail+"'
         " ); String msggg = "Your OTP for SSGK: " +DO_OTP ;
         Mail ma = new Mail();
         ma.secretMail(msggg, "OTP", mail);
         response.sendRedirect("otp.jsp");
      } else {
         response.sendRedirect("Data_Owner.jsp?Failed");
      }

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

Approve.jsp

```
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.ResultSet"%>
<%@page contentType="text/html"  pageEncoding="UTF-8"%>
<%
String mid=request.getParameter("mid");
String mail=request.getParameter("email");
Connection con = null;
   Statement st = null;
   Statement st1 = null;
   Connection conn =
   SQLconnection.getconnection(); Statement sto =
   conn.createStatement();
   st = conn.createStatement();

   try {
         int i = sto.executeUpdate("update member set statuss='Active' where id='" + mid + "'" );
        System.out.println("test print=="
         +mid); if (i != 0) {
            ResultSet rs = st.executeQuery(" SELECT * from member where id = '"+mid+"' ");
           if(rs.next()){
              String name=rs.getString("Name");
               String msggg= "Hi "+name+" Your account has been verified and Group service are activated
            "; Mail ma= new Mail();
         ma.secretMail(msggg,"Downloadkey",mail);
         String msg= "File Download Key:"+
         msggg;
            System.out.println("success");
            System.out.println("success");
            response.sendRedirect("sharing_request.jsp?Granted");
         } else {

            System.out.println("failed");
            response.sendRedirect("sharing_request.jsp?Failed");
             }
                 }
   } catch (Exception ex)
      {
      ex.printStackTrace();
   }
```

CHAPTER - 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In this project, a group key management protocol for file sharing on cloud storage. Public key are used by GKMP to guarantee the group key distribute fairly and resist attack from compromised vehicles or the cloud provider. We give detailed analysis of possible security attacks and corresponding defense, which demonstrates that GKMP is secure under weaker assumptions. Moreover we demonstrate the protocol exhibits less storage and computing complexity. Cloud computing has undeniably revolutionized the landscape of technology, offering unparalleled scalability, flexibility, and accessibility to businesses and individuals alike. With its ability to provide on-demand computing resources over the internet, cloud computing has significantly reduced the barriers to entry for companies of all sizes, enabling startups to compete on a level playing field with industry giants. Furthermore, the cost-effectiveness of cloud services, coupled with the elimination of infrastructure maintenance burdens, has allowed organizations to reallocate resources towards innovation and core business objectives. However, as the adoption of cloud computing continues to soar, concerns regarding data security, privacy, and regulatory compliance persist. Despite these challenges, the overall trajectory of cloud computing remains firmly upward, promising to reshape industries, drive digital transformation, and redefine the way we interact with technology in the years to come.

7.2 LIMITATIONS

- Security mechanism to conventional service oriented clouds, we obtain a security aware cloud and guarantee the privacy of data sharing on cloud storage.
- Quantity group numbers was n and the size of key
- GKMP just takes at most 190ms to process the secret key.
- The process of encryption key with five different group members GKMP just take at most 191ms.
- Resist attack
- Less storage and computing complexity.

7.3 FUTURE SCOPE OF PROJECT

In future to add forward and backward security in group key management may require some additions to our protocol. An efficient dynamic mechanism of group members remains as future work.

REFERENCES

1. https://link.springer.com/article/10.1007/s40747-020-00162-3
2. https://creativecommons.org/licenses/by/4.0
3. https://dl.acm.org/doi/10.1002/nem.2225#d77793155e1
4. https://chat.openai.com/