

A Project report on

**SAFE SHARING: ACCESS CONTROL FOR CLOUD
STORED DATA**

*Submitted in partial fulfillment of the requirements
for the award of the degree of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

G. AJAY KISHORE	(204G1A0506)
M. MOUNIKA	(204G1A0561)
T. HARSHA SRI	(204G1A0536)
B. BHAVANA	(204G1A0522)

Under the Guidance of

Mr. M. Narasimhulu M.Tech (Ph.D)

Assistant Professor



Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701

2023-2024

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi &

Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Certificate

This is to certify that the Project report entitled **SAFE SHARING: ACCESS CONTROL FOR CLOUD STORED DATA** is the bonafide work carried out by **G. Ajay Kishore, M. Mounika, T. Harsha Sri , B. Bhavana** bearing Roll Number **204G1A0506, 204G1A0561, 204G1A0536, 204G1A0522** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023 - 2024.

Project Guide

Mr. M. Narasimhulu M.Tech., (Ph.D)
Assistant Professor

Head of the Department

Mr. P.Veera Prakash M.Tech., (Ph.D)
Assistant Professor

Date:

External Examiner

Place: Rotarypuram

DECLARATION

We, Mr G. Ajay Kishore with reg no: 204G1A0506, Ms M. Mounika with reg no: 204G1A0561, Ms T. Harsha Sri with reg no: 204G1A0536, Ms B. Bhavana with reg no: 204G1A0522 students of **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, Rotarypuram, hereby declare that the dissertation entitled **“SAFE SHARING: ACCESS CONTROL FOR CLOUD STORED DATA”** embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of **Mr. M. Narasimhulu**, Assistant Professor, Department of CSE, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University or Institute for the award of any Degree or Diploma.

G. AJAY KISHORE

Reg no: 204G1A0506

M. MOUNIKA

Reg no: 204G1A0561

T. HARSHA SRI

Reg no: 204G1A0536

B. BHAVANA

Reg no: 204G1A0522

VISION & MISSION OF THE SRIT

Vision:

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

Mission:

- Continually enhance the quality of physical infrastructure and human resources to evolve in to a center of excellence in engineering education.
- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.
- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever-changing requirements of the industry.
- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

VISION & MISSION OF THE DEPARTMENT OF CSE

Vision:

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

Mission:

DM 1: Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mr. M. Narasimhulu, Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr. C. Lakshminatha Reddy, Assistant Professor** and **Mr. M. Narasimhulu, Assistant Professor**, Project Coordinators for their valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash, Assistant Professor & Head of the Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, We thank all other faculty and non- teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

Project Associates

204G1A0506

204G1A0561

204G1A0536

204G1A0522

ABSTRACT

The rapid expansion of cloud environments has brought about a significant challenge to secure data storage. This is a critical consideration for every user when decided to move the data online. To address this challenge, various solutions have been proposed, with two prominent approaches being Searchable Symmetric Encryption and Attribute-Based Encryption. SSE offers protection against both external and internal threats. It allows for efficient search capabilities while maintaining the confidentiality of the data. In an SSE technique, all data is usually encrypted using a single key. The entire encrypted database would need to be downloaded and re-encrypted with a new key if a user was to be revoked. Conversely, though, ABE offers a more granular approach to access control by encrypting data based on attributes and policies. This means that different users or groups can be granted different levels of access to the data based on their attributes.

Keywords:

Encryption, Cryptography, Access Control, Searchable Symmetric Encryption (SSE), Attribute-Based Encryption (ABE).

	CONTENTS	PAGE NO
	List of Figures	ix-x
	List of Tables	xi
	Abbreviations	xii
Chapter 1	Introduction	1-2
	1.1 Problem Statement	2
	1.2 Objectives	3
	1.3 Scope of Project	3
Chapter 2	Literature Survey	4-5
Chapter 3	System Analysis and Feasibility Study	6-14
	3.1 Architecture	6
	3.2 Software Development Life Cycle	7
	3.3 Feasibility Study	7
	3.3.1 Economic Feasibility	7
	3.3.2 Technical Feasibility	7
	3.3.3 Social Feasibility	9
Chapter 4	System Requirements Specification	10-11
	4.1 Functional Requirements	10
	4.2 Non-Functional Requirements	11
	4.3 Hardware & Software Requirements	11
Chapter 5	System Analysis and Design	12-17
	5.1 Input Design	12
	5.1.1 Objectives for Input Design	12
	5.2 Output Design	13
	5.2.1 Objectives of Output Design	13
	5.3 Modules	13
	5.3.1 User	13
	5.4 UML Diagrams	14
	5.5 Use Case Diagrams	15

	5.6 Class Diagrams	15
	5.7 Sequence Diagrams	15
	5.8 Collaboration Diagrams	16
	5.9 Activity Diagrams	16
	5.10 Component Diagrams	18
	5.11 Deployment Diagrams	19
Chapter 6	System Installation	20-29
	6.1 Software Environment	20
	6.1.1 Java Technology	20
	6.1.2 The Java Programming Language	20
	6.1.3 The Java Platform	20
	6.1.4 What Can Java Technology Do	23
	6.1.5 JDBC	25
	6.1.6 ODBC	25
	6.1.7 Eclipse	28
	6.1.8 SQL Level API	28
Chapter 7	Output Screenshots with Description	30-33
	CONCLUSION	34
	REFERENCES	35-36
	PUBLICATION	
	CERTIFICATE	

LIST OF FIGURES

Fig. No.	Description	Page No.
3.1	System Architecture	6
3.2	Waterfall Model	7
5.5.1	Use Case Diagrams	13
5.6.1	Class Diagrams	14
5.7.1	Sequence Diagrams	15
5.8.1	Collaboration Diagrams	15
5.9.1	Activity Diagrams	16
5.10.1	Component Diagrams	16
5.11.1	Deployment Diagrams	17
6.1	Java Working Process	18
6.2	Java Platform	18
6.3	Java Compilation	18
6.4	Eclipse	19
7.1	Initial Page	27
7.2	Registration Page	27
7.3	User Login Page	28
7.4	User Home Page	28
7.5	Upload Page	29
7.6	View Uploaded Page	29
7.7	Share Files Page	30
7.8	Search Files Page	30
7.9	User Request Files Page	31
7.10	Download Files Page	31

LIST OF TABLES

Table No.	Title	Page No.
7.1	Test Cases	33

LIST OF ABBREVIATIONS

SSE	Searchable Symmetric Encryption
ABE	Attribute Based Encryption
CSP	Cloud Service Provider
RBAC	Role Based Access Control
SRS	System Requirements Specification
TCP	Transmission Control Protocol
UML	Unified Modeling Language
IDE	Integrated Development Environment
SDLC	Software Development Life Cycle

CHAPTER 1

INTRODUCTION

People may rapidly and simply execute crucial tasks with their data in cloud computing, such as locating, transferring, and conserving it. However, maintaining the security of the data is a challenge. This is because the information is maintained by a different organization, and poorly protected data carries the highest risks.

The last several years have seen such rapid development in cloud computing that almost everyone's everyday life is now significantly impacted by it. The cloud is currently used on a daily basis by both large corporations and regular internet users. Many users are still hesitant to outsource their personal information, though, because cloud services are housed and managed by questionable third parties, making the contents susceptible to internal attacks.

Major players in the business as well as researchers have looked at attribute-based and symmetric searchable encryption as potential solutions for this reason. Before sending their files to the Cloud Service Provider (CSP), users in an SSE system encrypt them locally. As a result, the CSP that does not have the encryption key cannot obtain any relevant information about the users' data. The ability to do a direct keyword search on encrypted data is the most exciting feature of SSE, though. Unfortunately, user revocation is not supported by SSE systems, which is a major problem for cloud-based apps. Thus, eliminating a user corresponds to download the entire database and again encrypting it using a new key.

An alternative approach that functions in cloud-based applications is ABE. A master public key is used to encrypt every file in ABE schemes; however, Unlike conventional public key cryptosystems, which use the ciphertext that is generated is limited by a policy. Every user also has a secret key which is unique and associated with the users attributes. As a result a file can only be unlocked if and when the user's attributes align with the ciphertext's policy. However, encrypting vast amounts of data with an asymmetric encryption method is not particularly effective.

Additionally, the solution must ensure secure authentication and authorization mechanisms to verify the identities of users and enforce access policies effectively. It should also incorporate encryption techniques to protect data during transit and at rest, preventing unauthorized access even if the data is compromised.

Ultimately, the goal is to establish a comprehensive access control solution that balances security with usability, enabling secure data sharing and collaboration in cloud environments while mitigating the risks associated with unauthorized access and data exposure.

1.1 Problem Statement

As organizations increasingly rely on cloud storage solutions for data sharing and collaboration, ensuring the security and privacy of shared data becomes paramount. However, traditional access control mechanisms often fall short in adequately safeguarding data from unauthorized access or leakage. The challenge lies in implementing a robust access control system that maintains data confidentiality, integrity, and availability while enabling seamless collaboration among authorized users.

Specifically, the problem entails developing an access control framework for cloud-shared data that addresses various security threats, including insider threats, unauthorized access attempts, and data breaches. This framework should incorporate fine-grained access control policies based on user roles, privileges, and attributes to restrict data access to authorized individuals or groups. Moreover, it should support dynamic access control adjustments to accommodate evolving user permissions and organizational requirements.

- The increase of data breaches in cloud computing puts all users at risk of business problems, highlighting the need to improve security measures.
- To mitigate the growing risks of data threats faced by users in cloud Environment, proactive measures are essential.

1.2 Objectives

To accomplish the project's purpose, the following particular objectives have been established.

- To create an user interface and implementing searchable encryption on the text file uploaded by the user.
- To implement the access control scheme and addressing the problem of revocation by using cryptographic techniques.

1.3 Scope of the Project

The following are the boundaries that have established in the proposed system which defines scope.

- In proposed system, we use these two cryptographic techniques that secure the data storage in cloud-based environments, to design a hybrid encryption scheme based on ABE and SSE in such a way that we utilize the best out of both of them.
- The data can be encrypted into cipher text before moving into cloud by using symmetric encryption.
- The revocation mechanism and an access control bounded by policy, that ensures data security

CHAPTER 2

LITERATURE SURVEY

Michalas A and A. Bakas presented a novel technique that enables data owners to link specific policies to specific areas of their cipher texts. The scheme is based on existing symmetric primitives. They combined an in-depth simulation-based security study with an experimental evaluation that demonstrates our scheme's efficacy to demonstrate the accuracy of our methodology [1].

A. Sharma claims in J. Bethencourt study that using a trusted server to store data and handle access control is the only way to enforce such regulations. In their system, a party encrypting data establishes a policy for who can decrypt, and attributes are used to characterize a user's credentials. Consequently, techniques like role-based access control (RBAC) [2]

The safe and effective oblivious storage systems described in the paper by Y. XU and W. Cui concentrate on making use of all available network bandwidth to provide concurrent access through a reliable proxy. However, the performance is limited by the network's latency and bandwidth because the proxy uses the network to carry out a common ORAM protocol. Furthermore, in such proxy environments, several crucial elements like access control and security against active adversaries have not been well investigated [3].

The study by R. Dowsley shows how to create a hybrid encryption scheme that combines SSE and ABE while taking advantage of their respective benefits. Unlike many other methods, we build a revocation process that is based only on SGX's capability and is totally independent of the ABE scheme [4].

The concept of backward privacy for searchable encryption is examined for the first time in the study by R. Bost and B.Minaud. Following the theoretical definitions of several flavors of backward privacy, we propose multiple strategies with varied efficiency trade-offs that achieve both forward and backward privacy. Importantly, our constructs depend on primitives like

puncturable encryption schemes and limited pseudo-random functions[5].

The traditional system says that the cloud service provider will encrypt the user's data and stores in it which is not acceptable by the user because the cloud services can access their data. So to address this problem , they have proposed symmetric searchable encryption where data encrypted before sending into cloud. Here, only user can encrypt and decrypt the document by using encrypted Keyword. The Encrypted Keyword can be generated by the secret key of the user.

The system says that In several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call ciphertext-policy attribute-based encryption.

CHAPTER 3

SYSTEM ANALYSIS & FEASIBILITY STUDY

System architecture focuses on designing the overall structure of a system, while system analysis involves understanding the current state of the system and proposing improvements or enhancements to meet the project's objectives. Both are essential for ensuring the success of a project and delivering a system that meets the needs of its users.

3.1 Architecture:

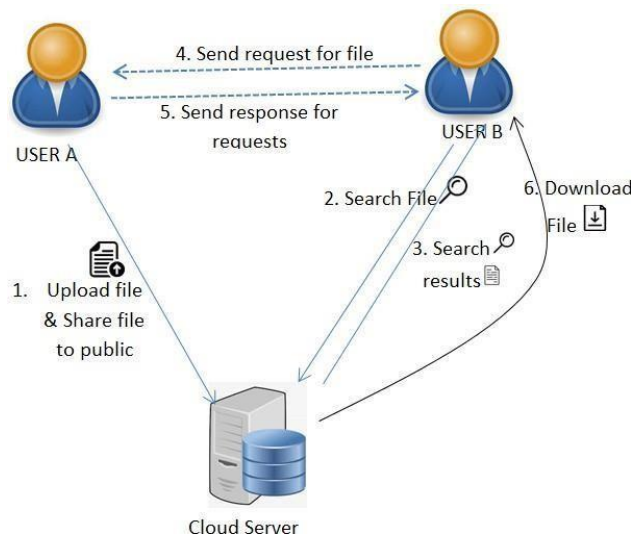


Fig. 3.1.1: System Architecture

3.2 SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.

- **Requirement Gathering and analysis** – all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – the requirement specifications from first phase are studied in this phase and the system design is prepared. This system

design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

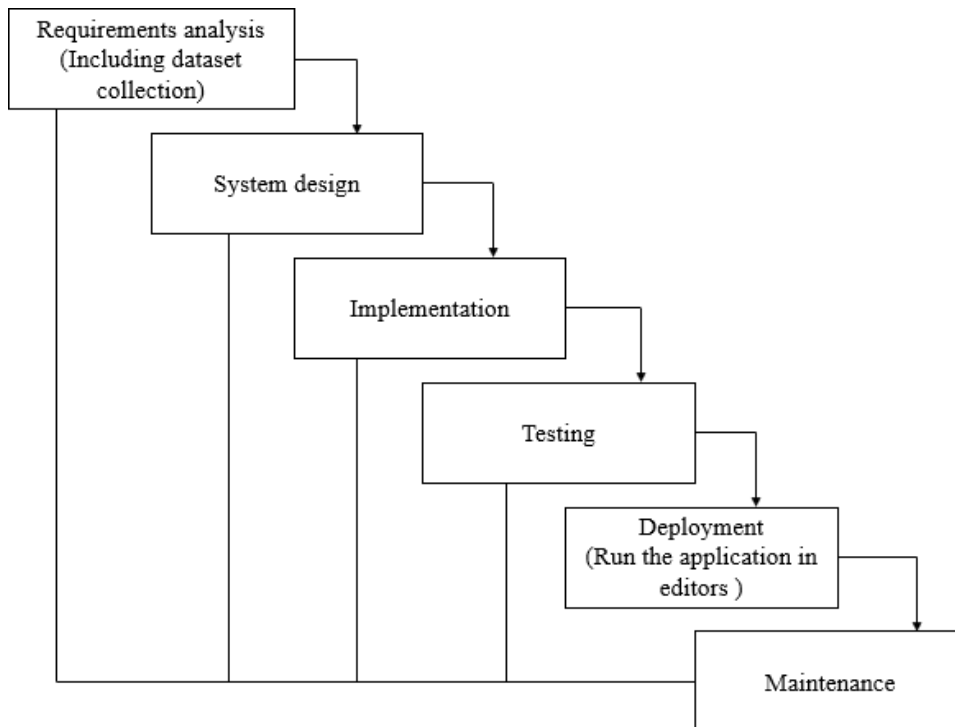


Fig. 3.2.1: Waterfall Model

- **Implementation** – with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance

the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

3.2 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

3.2.1 Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2.2 Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.2.3 Social feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

SYSTEM REQUIREMENTS SPECIFICATION

4.1 Functional Requirements

Such requirements describe system behavior under specific conditions and include the product features and functions which web & app developers must add to the solution. Such requirements should be precise both for the development team and stakeholders.

The list of examples of functional requirements includes:

- Business Rules
- Transaction corrections, adjustments, and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data

4.2 Non-Functional Requirements

Hence few things to be noted before Mobile Application Development are

- Unlike laptops/Desktop the resources available on Mobile devices are very less like processor, speed, screen dimension etc.
- Scalability of device screen is different for different mobiles.
- Network condition may vary.
- Multitasking capability and Memory available.
- Different version of OS and backward compatibility etc.

Moving on let me List out few Key types of NFR that needs to be taken care of

- Performance
- Scalability
- Responsiveness
- Use-ability
- reliability
- Security
- Documentation
- Availability

4.3 HARDWARE & SOFTWARE REQUIREMENTS

Software and hardware requirements are essential aspects of system development, whether it's for a software application, a website, or any other technology-driven project. These requirements outline what software and hardware components are necessary to effectively run the system.

4.3.1 Hardware System Configurations :-

- Processor - I3/Intel Processor
- RAM - 4GB (min)
- Hard Disk - 500GB

4.3.2 Software System Configurations :-

- Operating System : Windows 7/8/10
- Application Server : Tomcat 7.0
- Front End : HTML, JSP
- Scripts : JavaScript.
- Server side Script : Java Server Pages.
- Database : My SQL 6.0
- Database Connectivity : JDBC

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

5.1 Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output.

Well-designed input forms and screens have following properties –

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding –
 - What are the inputs needed for the system?
 - How end users respond to different elements of forms and screens.

5.1.1 Objectives for Input Design:

The objectives of input design are

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

5.2 Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

5.2.1 Objectives of Output Design:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

5.3 MODULES:

MODULES :

5.3.1 User :

- **Register:** user can enter the details and he can register.
- **Login:** user can login with his valid credentials. If user enter invalid credentials then it can be redirect into login page. If user enters valid credentials then it can be redirect into user home.
- **Upload:** Here, User can Upload the files. While uploading the file it can be store into encrypted format and generates searchable keywords.
- **View Files :** The user view the uploaded files and share files to other users.
- **Search:** user can search for a file based on keywords. If file has been found send request to file uploaded user
- **View request :** In this user view the request from other users for their file then user can accept/ Reject the request

- **Status** : The user view the file requested status, i.e. pending and Accepted.
- **Download**: user can download the file if his request is accepted. Here Encryption file converted into decryption format (original File) can be Download.

5.4 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

5.5 USE CASE DIAGRAMS:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

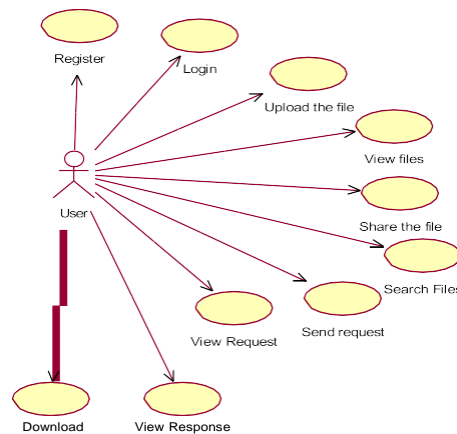


Fig. 5.5.1: Use Case Diagram

5.6 CLASS DIAGRAMS:

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

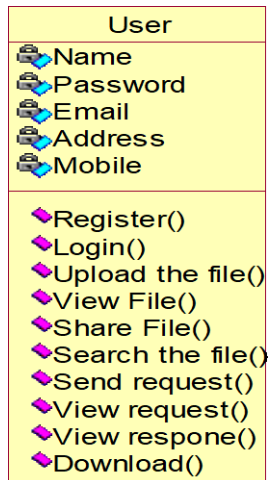


Fig. 5.6.1: Class Diagram

5.7 SEQUENCE DIAGRAMS:

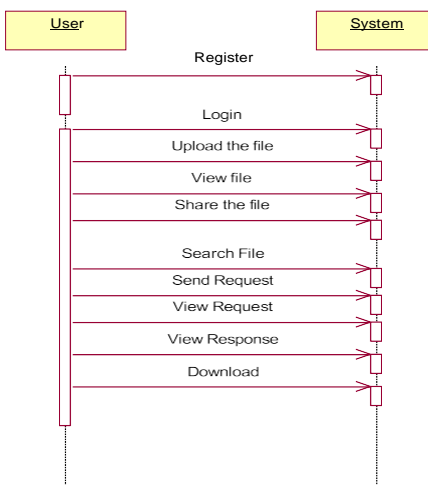


Fig. 5.7.1 Sequence Diagram

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

5.8 COLLABORATION DIAGRAMS:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

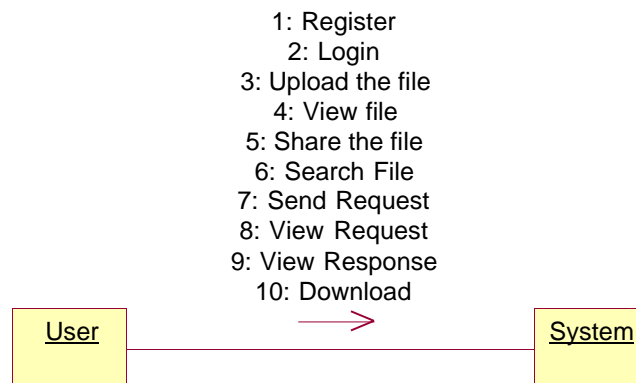


Fig. 5.8.1: Collaboration Diagram

5.9 ACTIVITY DIAGRAMS:

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.

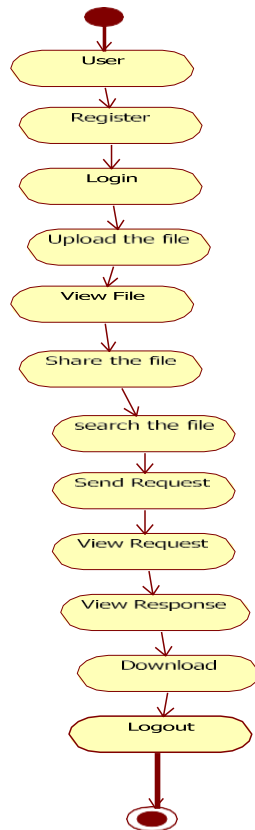


Fig. 5.9.1: Activity Diagram

5.10 COMPONENT DIAGRAMS:

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries etc. So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.



Fig. 5.10.1: Component Diagram

5.11 DEPLOYMENT DIAGRAMS:

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.



Fig. 5.11.1: Deployment Diagram

CHAPTER 6

SYSTEM INSTALLATION

6.1 Software Environment

Software installation involves deploying and configuring software on a computing system, while the environment encompasses the broader context in which the software operates, including hardware, software dependencies, configuration settings, and security measures. Understanding both installation procedures and the environment is crucial for effectively deploying and maintaining software systems.

6.1.1 Java Technology

Java technology is both a programming language and a platform.

6.1.2 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. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

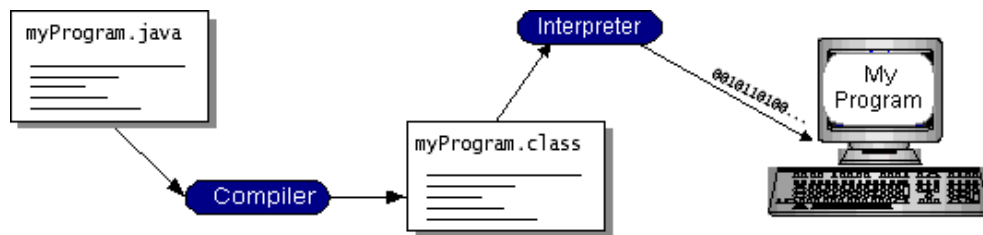


Fig. 6.1: Java Working Process

6.1.3 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 other hardware-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.

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.

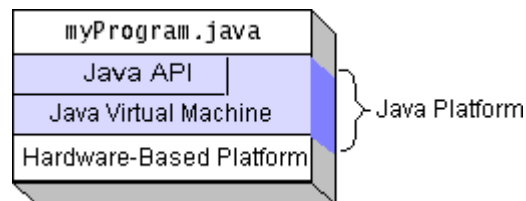


Fig. 6.2 Java Platform

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

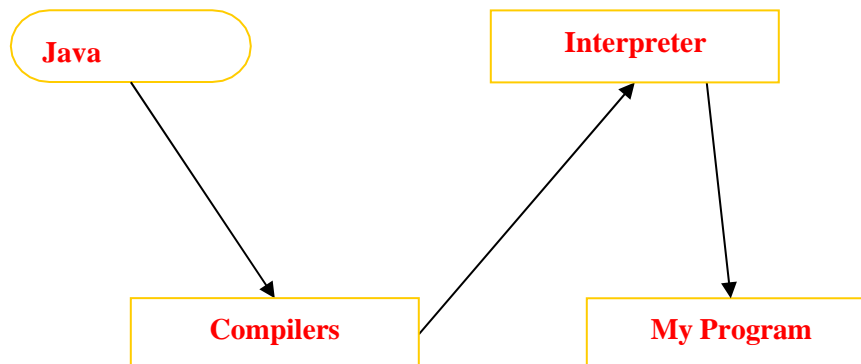


Fig. 6.3 Java Compilation

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make “write once, run anywhere” possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

6.1.4 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.

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.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

Get started quickly: Although the Java programming language is a powerful

object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.

Write less code: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

Write better code: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.

Develop programs more quickly: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

Write once, run anywhere: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

Distribute software more easily: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

6.1.5 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of

“plug-in” database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

6.1.6 ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas

the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources. From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives

you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

6.1.7 Eclipse

The Eclipse download requires about 300 MB of disk space; keep it on your machine, in case you need to re-install Eclipse. When installed, Eclipse requires an additional 330 MB of disk space.

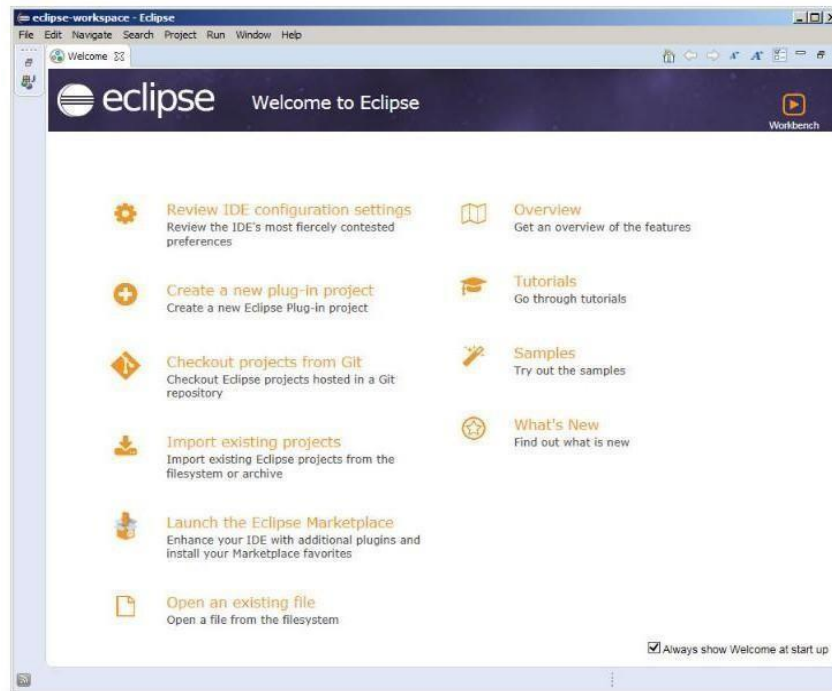


Fig. 6.4: Eclipse WorkSpace

6.1.8 SQL Level API

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

1. JDBC must be implemental on top of common database interfaces

The JDBC SQL API must “sit” on top of other common SQL level APIs.

This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

2. Provide a Java interface that is consistent with the rest of the Java system

Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

3. Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

4. Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

5. Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

CHAPTER 7

OUTPUT SCREEN SHOTS WITH DESCRIPTION.

Home: This is the Initial Page of our Project



Fig. 7.1: Home Page

User Registration: Registration

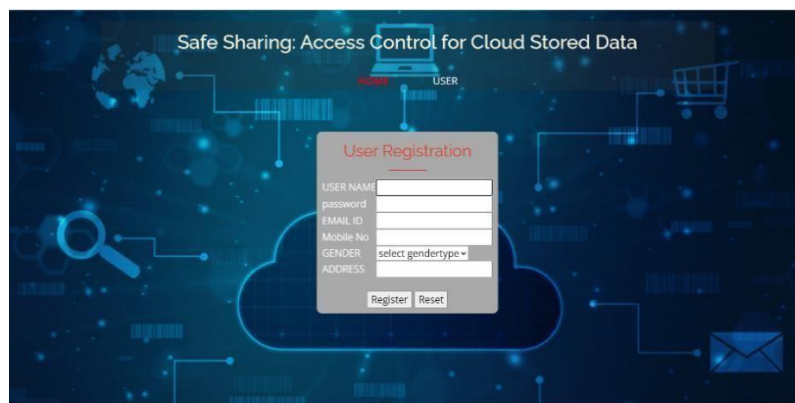


Fig. 7.2: Registration

User Login : This is the Login Page

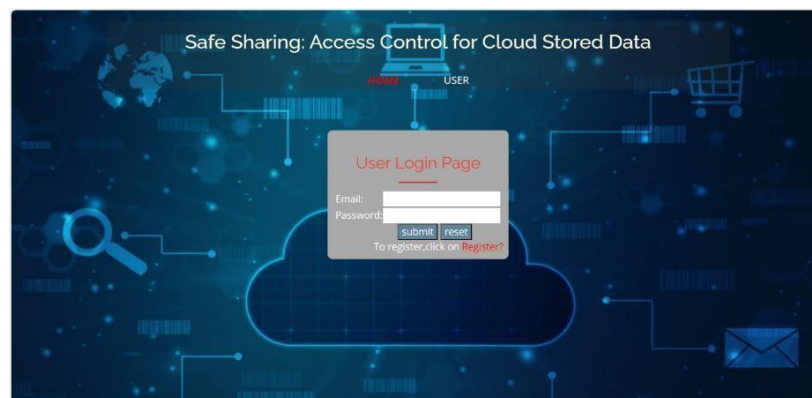


Fig. 7.3: Login Page

User Home: This is the Home Page



Fig. 7.4: Home Page

Upload: User Uploading Files



Fig. 7.5: Files Upload

View: User Viewing his/her Files



Fig. 7.6: Viewing the Files

File Sharing : Files to be Shared by the User

Safe Sharing: Access Control for Cloud Stored Data

HOME

LOGOUT

UPLOAD FILES

VIEWFILES

NEW FILE

SEARCH FILES

REQUEST

STATUS

id	File Name	Uploaded File	Download	Share the file
20file1	har1.txt	Download	Shared	
23file12	har1.txt	Download	Shared	
24file23	code.txt	Download	Shared	
25file40	code.txt	Download	Shared	
27file14	code.txt	Download	Shared	
28file12	code.txt	Download	Shared	
29file12	code.txt	Download	Shared	
30file15	code.txt	Download	Shared	
32java	code.txt	Download	File Sharing	
33java	code.txt	Download	Shared	
35kk	code.txt	Download	Shared	
36l	code.txt	Download	Shared	
37file1	code.txt	Download	Shared	
38nazeema	Java Files.txt	Download	File Sharing	
39file21	code.txt	Download	Shared	
41file1	code.txt	Download	Shared	
42java	code.txt	Download	Shared	
43java	code.txt	Download	Shared	

Fig. 7.7: User Sharing Files

File Search: User Searching the Files by a Keyword



Fig. 7.8: Searching the Files

View Searched Files: User can find the searched files and requesting for its Access



Owner Id	File Id	File Name	Action
24	18	Java	Request

Fig. 7.9: Searched Files

User Response: User can accept the request

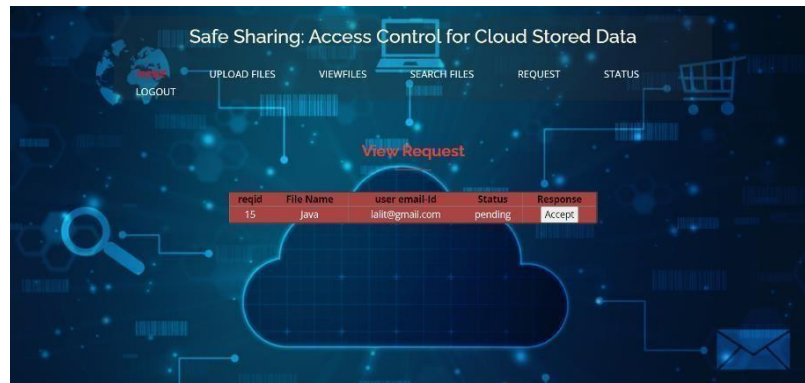


Fig. 7.10: Accepting the Request

Download: User downloading the Files

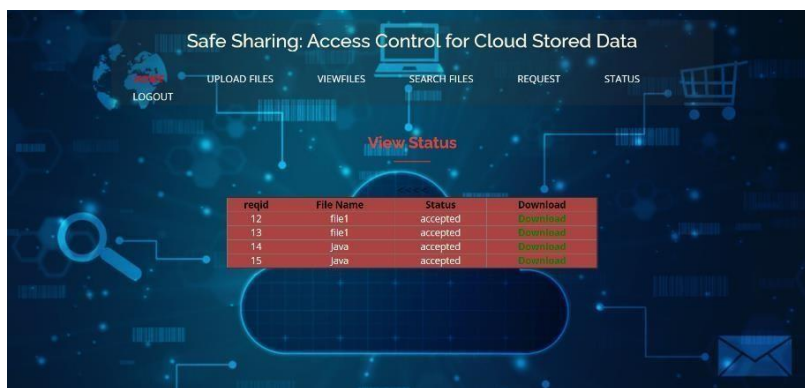


Fig. 7.11: User downloading the files

TEST CASES

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	User login	Valid/invalid credentials	Login success/ login failed	Login success/ login failed	P
2	User register	User details	If email not exist registration done else email already exists	If email not exist registration done else email already exists	P

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION

In this paper, we proposed The Cloud we Share, a hybrid encryption scheme based on SSE and ABE. Our construction allows a data owner to share her data in a privacy-preserving way and manage the access rights of the rest of the users

Future Scope: In future we can implement to More security and provide Email Authentication.

REFERENCES

- [1] S. Agrawal and M. Chase, “[FAME: Fast attribute-based message encryption](#),” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017, pp. 665–682.
- [2] G. Amjad, S. Kamara, and T. Moataz, “[Forward and backward private searchable encryption with SGX](#),” in Proc. 12th Eur. Workshop Syst. Secur. (EuroSec). New York, NY, USA: Association for Computing Machinery, 2019.
- [3] A. Bakas and A. Michalas, “[Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX](#),” in Security and Privacy in Communication Networks, S. Chen, K.-K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, Eds. Cham, Switzerland: Springer, 2019, pp. 472–486.
- [4] A. Bakas and A. Michalas, “[Multi-client symmetric searchable encryption with forward privacy](#),” Cryptol. ePrint Arch., Tampere Univ., Tampere, Finland, Tech. Rep. 2019/813, 2019. [Online]. Available: <https://eprint.iacr.org/2019/813>
- [5] A. Bakas and A. Michalas, “[Power range: Forward private multi-client symmetric searchable encryption with range queries support](#),” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jul. 2020, pp. 1–7.
- [6] J. Bethencourt, A. Sahai, and B. Waters, “[Ciphertext-policy attribute-based encryption](#),” in Proc. IEEE Symp. Secur. Privacy (SP). Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334.
- [7] A. Boldyreva, V. Goyal, and V. Kumar, “[Identity-based encryption with efficient revocation](#),” in Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2008, pp. 417–426.
- [8] D. Boneh, X. Boyen, and E.-J. Goh, “[Hierarchical identity based encryption with constant size ciphertext](#),” in Advances in Cryptology—EUROCRYPT, R. Cramer, Ed. Berlin, Germany: Springer, 2005, pp. 440–456.
- [9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “[Public key encryption with keyword search](#),” in Proc. Int. Conf. Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2004, pp. 506–522.
- [10] R. Bost, “[σ οφος: Forward secure searchable encryption](#),” in Proc. ACM

SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1143–1154.

[11] R. Bost, B. Minaud, and O. Ohrimenko, “[“Forward and backward private searchable encryption from constrained cryptographic primitives,”](#) in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017, pp. 1465–1482.

[12] V. Boyko, “[“On the security properties of OAEP as an all-or-nothing transform,”](#) in Advances Cryptology— CRYPTO, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 503–518,

[13] V. Costan and S. Devadas, “[“Intel SGX explained,”](#) Cryptol. ePrint Arch., Intel, Mountain View, CA, USA, Tech. Rep. 2016/086, 2016.

[14] R. Dowsley, A. Michalas, M. Nagel, and N. Paladi, “[“A survey on design and implementation of protected searchable data in the cloud,”](#) Comput. Sci. Rev., vol. 26, pp. 17–30, Nov. 2017.

[15] M. Etemad, A. Küpçü, C. Papamanthou, and D. Evans, “[“Efficient dynamic searchable encryption with forward privacy,”](#) Proc. Privacy Enhancing Technol., vol. 2018, no. 1, pp. 5–20, Jan. 2018.

[16] B. Fisch, D. Vinayagamurthy, D. Boneh, and S. Gorbunov, “[“IRON: Functional encryption using Intel SGX,”](#) in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Oct. 2017, pp. 765–782.

[17] B. Fuhry, R. Bahmani, F. Brasser, F. Hahn, F. Kerschbaum, and A.-R. Sadeghi, “[“HardIDX: Practical and secure index with SGX,”](#) in Data and Applications Security and Privacy, G. Livraga and S. Zhu, Eds. Cham, Switzerland: Springer, 2017, pp. 386–408,

[18] S. Lee, M. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, “[“Inferring fine-grained control low inside SGX enclaves with branch shadowing,”](#) in Proc. 26th USENIX Secur. Symp., Victoria, BC, Canada, Aug. 2017, pp. 557–574.

[19] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, “[“Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list,”](#) in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur. Cham, Switzerland: Springer, Jul. 2018, pp. 516–534.

Safe Sharing: Access Control for Cloud Stored Data

Narasimhulu Malavathula,^{1, a)} Ajay Kishore Gattu, Mounika Meenuga,
Harsha Sri Talanki, Bhavana Bommineni^{2,3,4,5 b)}

¹Assistant Professor, Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, Anantapur, India.

^{2,3,4,5} Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology

^{a)}narasimhulu.cse@srit.ac.in,

^{b)}204g1a0506@srit.ac.in

^{c)}204g1a0561@srit.ac.in

^{d)}204g1a0536@srit.ac.in

^{e)}204g1a0522@srit.ac.in

Abstract. The rapid expansion of cloud environments has brought about a significant challenge to secure data storage. This is a critical consideration for every user when decided to move the data online. To address this challenge, various solutions have been proposed, with two prominent approaches being Searchable Symmetric Encryption and Attribute-Based Encryption. SSE offers protection against both external and internal threats. It allows for efficient search capabilities while maintaining the confidentiality of the data. In an SSE technique, all data is usually encrypted using a single key. The entire encrypted database would need to be downloaded and re-encrypted with a new key if a user was to be revoked. Conversely, though, ABE offers a more granular approach to access control by encrypting data based on attributes and policies. This means that different users or groups can be granted different levels of access to the data based on their attributes.

Keywords: Encryption, Cryptography, Access Control, Searchable Symmetric Encryption (SSE), Attribute-Based Encryption(ABE)

INTRODUCTION

People may rapidly and simply execute crucial tasks with their data in cloud computing, such as locating, transferring, and conserving it. However, maintaining the security of the data is a challenge. This is because the information is maintained by a different organization, and poorly protected data carries the highest risks.

The last several years have seen such rapid development in cloud computing that almost everyone's everyday life is now significantly impacted by it. The cloud is currently used on a daily basis by both large corporations and regular internet users. Many individuals are yet reluctant to delegate their personal data due to the fact that cloud services are hosted and overseen by dubious third parties, rendering the data vulnerable to internal breaches.

Major players in the business as well as researchers have looked at attribute-based and symmetric searchable encryption as potential solutions for this reason. Before transmitting their files to the Cloud Service Provider (CSP), individuals participating in SSE employ local encryption. Consequently, the CSP, lacking the encryption key, is unable to access any pertinent details regarding the user data. The ability to do a direct keyword search on encrypted data is the most exciting feature of SSE, though. Unfortunately, user revocation is not supported by SSE systems, which is a major problem for cloud-based apps. Thus, eliminating a user corresponds to download the complete database and again encrypting it using a new key.

An alternative approach that functions in applications that utilize cloud technology is ABE. A master public key is employed to encrypt all files in ABE schemes; nevertheless, in contrast to traditional public key cryptosystems, which use the ciphertext that is generated is limited by a policy. Every user also has a secret key which is unique and associated with the user's attributes. As a result a file can be unlocked only if and when the characteristics possessed by the user align with the ciphertext's policy. However, encrypting vast amounts of data with an asymmetric encryption method is not particularly effective.

LITERATURE SURVEY

A. Michalas and A. Bakas presented a novel technique that enables data owners to link specific policies to specific areas of their cipher texts. The scheme is based on existing symmetric primitives. They combined an in-depth a security study that employs simulation-based methods and includes an experimental evaluation is conducted to showcase the effectiveness of our scheme, thereby demonstrating the precision of our methodology[1].

A. Sharma claims in J. Bethencourt study that using a reliable server for data storage and access control is the only way to enforce such regulations. In their system, a party encrypting data create a policy for who can decrypt, and attributes are used to characterize a user's credentials. Consequently, techniques like role-based access control (RBAC) [2]

The primary objective of this study is to emphasize and give attention to an encryption scheme called efficient revocable Ciphertext-Policy Attribute-Based Encryption. This particular encryption scheme is designed to allow the revocation of access to encrypted data based on specific attributes associated with the intended recipient. This means that if a user's attributes change, they can be revoked access to the encrypted data, ensuring security and privacy. The scheme focuses on enhancing the efficiency of the revocation process, ensuring that it can be done in a timely and effective manner. By utilizing this encryption scheme, organizations and individuals can have more control over who can access their encrypted data, enhancing overall data security. [3].

The study by R. Dowsley shows how to create a hybrid encryption scheme that combines SSE and ABE while taking advantage of their respective benefits. Unlike many other methods, we build a revocation process that is based only on SGX's capability and is totally independent of the ABE scheme [4].

The concept of backward privacy for searchable encryption is examined for the first time in the study by R. Bost and B.Minaud. Following the theoretical definitions of several flavours of backward privacy, we propose multiple strategies with varied efficiency trade-offs that achieve both forward and backward privacy. Importantly, our constructs depend on primitives like puncturable encryption schemes and limited pseudo-random functions[5].

PROBLEM DEFINITION

Due to the rise in data breaches in cloud computing, all users are vulnerable to business issues. Proactive approaches are crucial in mitigating the increasing danger of threads that users encounter in cloud environments, while also emphasizing the need for improved security controls. The major objective is to use various encryption algorithms, such as SSE and ABE, at untrusted clouds to create advanced protection for user assets.

PROPOSED SYSTEM

The proposed approach is compatible with deployment models for private, communal, or hybrid clouds. The suggested framework consists of the sensitive data in the cloud is encrypted using a hybrid encryption approach. Before the data is transferred to the cloud, it will be encrypted with a public key. We have specific access that only specific individuals can access the cloud.

An authorized user has the ability to view and edit cloud data information. A user can extract a certain block of code using the SSE Algorithm and use it to decrypt a particular file. Not all users of the ABE Scheme will be able to access data; authentication will only be granted to specific users.

BLOCK DIAGRAM

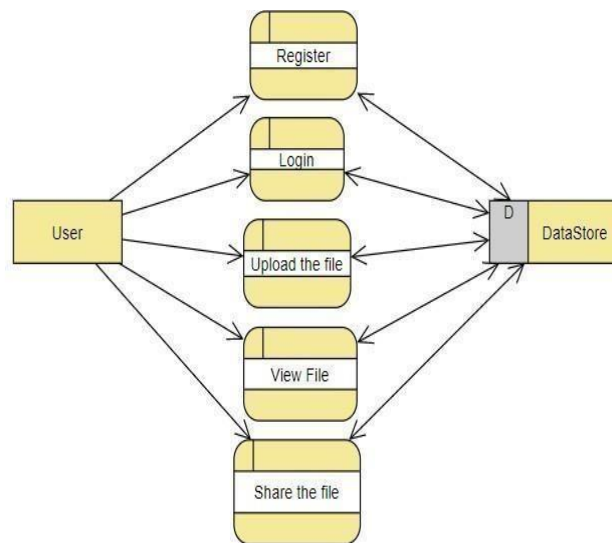


FIGURE 1: Block Diagram

The user will first register using his email address and password in the diagram. Following page login, he or she will upload a file and be able to view it or retrieve it by using a term as a key. By employing that keyword, the user can so share the file with others. The credentials are crucial for the system in the main.

Architecture

The flow diagram that follows provides an explanation of how the system operates. The steps that make up the overall process are as follows.

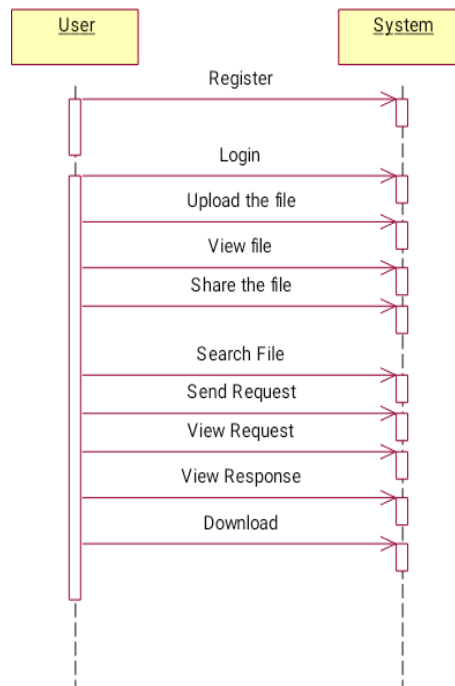


FIGURE 2: Flow Diagram.

SYSTEM IMPLEMENTATION

The methodology consists of a set of steps that must be followed in a specific order for the process to be completed. Since the waterfall model is used in the methodology, the suggested system meets the requirements by creating planning in a way that ensures steps are completed in a methodical manner

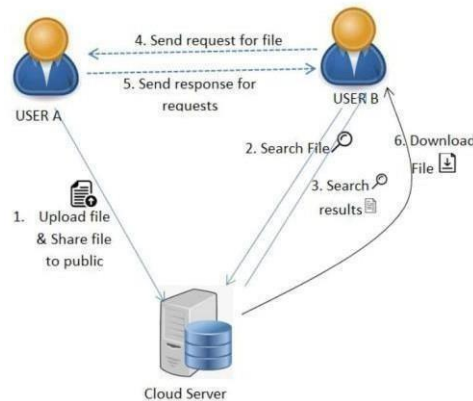


FIGURE 3: Architecture

Register

After entering his information, the user can register.

Login

Using legitimate credentials, the user can log in. The user may be redirected to the login page if they provide invalid credentials. The user may be sent to their home if they submit proper credentials.

Upload

The user may upload files here. The file can be generated with searchable keywords and stored in an encrypted format during the upload process.

View Files

The user can share files with other users and view files that have been submitted.

Search

Using keywords, users can look for files. Send a request to the file uploaded user if the file has been located.

View request

This feature allows the user to see requests made by other users for their files, which they can either accept or reject.

Status

The feature allows the user to check the requested file's pending and accepted states.

Download

Should the user's request be approved, he can get the file. The original encryption file, which has been transformed into a decryption format, can be downloaded here

FIGURE 4: Use Case Diagram

EFFICIENCY OF ALGORITHMS

It's difficult to provide an exact percentage of the security offered by Attribute-Based Encryption (ABE) and Symmetric Searchable Encryption (SSE) as it is reliable on various factors such as the strength of encryption algorithms, key management practices, and implementation details. However, both ABE and SSE can provide a high level of security, typically well above 90% when implemented correctly and used in appropriate scenarios.

RESULTS & DISCUSSION

The user will be the only source of dependency for the proposed system. Given that we are using a webpage to demonstrate how the system functions. The outputs from the registration stage to the file retrieval and sharing stage are displayed in the images below. Users can only share their files with other users if the keyword matches the index number. Users can only grant access to other users to other files if they send a request. only the user can determine whether to grant access.

Home : this is the intial page of the project



FIGURE 5: Home Page

User Registration:



The registration page features a dark blue background with a grid of icons including a globe, a magnifying glass, a shopping cart, and an envelope. A central white cloud contains a registration form. The form has a title 'User Registration' and fields for 'USER NAME', 'password', 'EMAIL ID', 'Mobile No', 'GENDER' (with a dropdown menu), and 'ADDRESS'. At the bottom of the form are 'Register' and 'Reset' buttons. Above the form, the text 'Safe Sharing: Access Control for Cloud Stored Data' is displayed, along with a 'USER' label and a laptop icon.

Safe Sharing: Access Control for Cloud Stored Data

USER

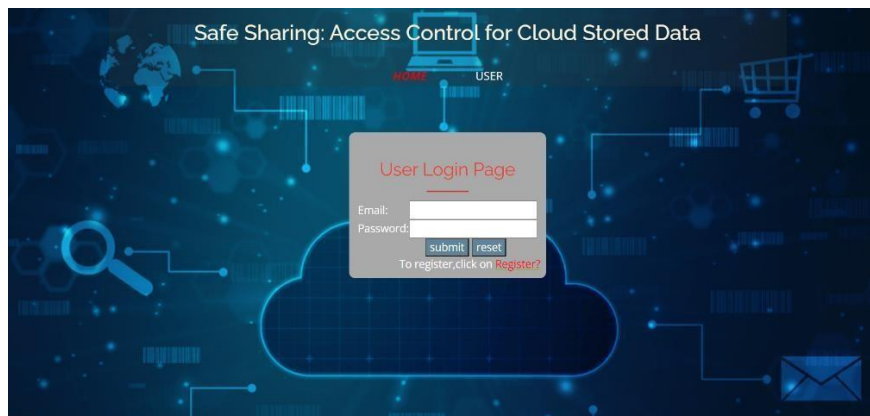
User Registration

USER NAME
password
EMAIL ID
Mobile No
GENDER: select gendertype
ADDRESS

Register Reset

FIGURE 6: Registration Page

User login



The login page has the same background and layout as the registration page. The central white cloud contains a login form titled 'User Login Page'. It has fields for 'Email:' and 'Password:', followed by 'Submit' and 'reset' buttons. Below the form, it says 'To register,click on Register?'. The header text and icons are identical to the registration page.

Safe Sharing: Access Control for Cloud Stored Data

USER

User Login Page

Email:
Password:

Submit reset

To register,click on Register?

FIGURE 7: Login Page

User Files Upload



The upload page features the same background and layout. The central white cloud contains a file upload form titled 'Upload your files here...'. It has fields for 'File Name:' (with 'Java' entered), 'Attach File:' (with a 'Choose File' button and 'Java Files.txt' selected), and 'Index Value:' (with '333' entered). At the bottom of the form are 'upload' and 'clear' buttons. The header text is identical, but the navigation bar includes 'UPLOAD FILES', 'VIEW FILES', 'SEARCH FILES', 'REQUEST', 'STATUS', and 'LOGOUT' (with a red 'Logout' button).

Safe Sharing: Access Control for Cloud Stored Data

Logout UPLOAD FILES VIEW FILES SEARCH FILES REQUEST STATUS

Upload your files here...

File Name: Java
Attach File: Choose File Java Files.txt
Index Value: 333

upload clear

FIGURE 8: User Uploading Files

View Files

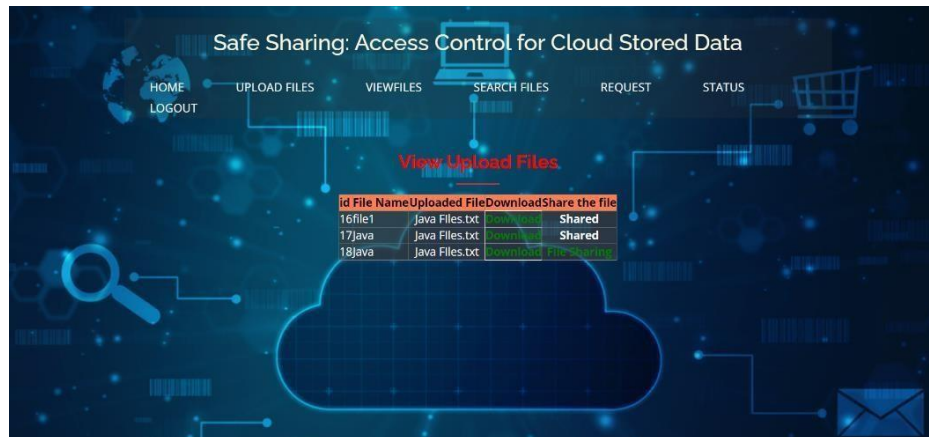


FIGURE 9: User Viewing Files

Search Files

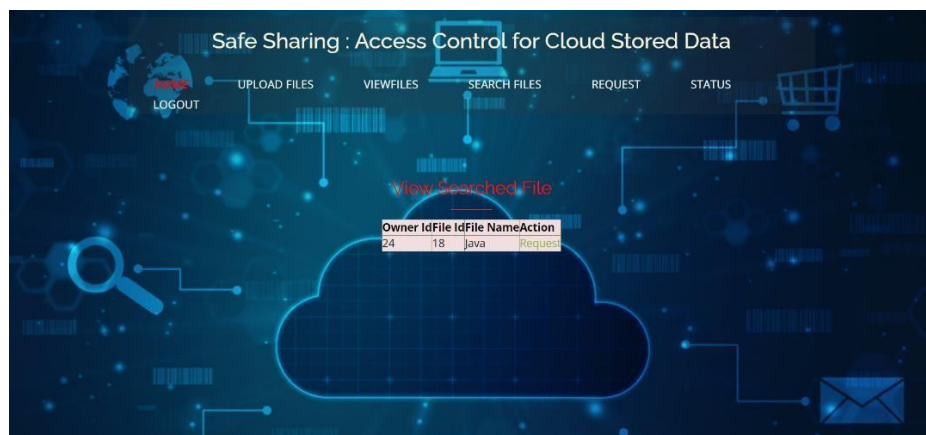


FIGURE 10: User Searching Files

Request Files



FIGURE 11: Request Acceptance

Status

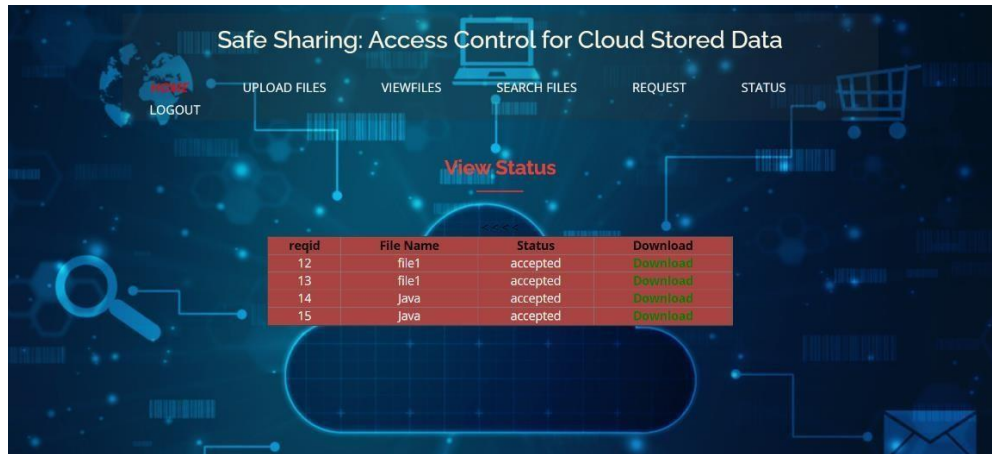


FIGURE 12: Status of the Page

CONCLUSION

Our project combines the strengths of both Symmetric Searchable Encryption (SSE) and Attribute-Based Encryption (ABE) to ensure secure data storage in cloud environments. SSE provides efficient search capabilities and protection against attacks, while ABE offers fine grained access control based on attributes and policies. By combining these two techniques, we aim to create a web application that meets the diverse security needs of our users

FUTURE SCOPE

In future we can implement to more security and provide two step authentication.

REFERENCES

- [1]. A. Bakas and A. Michalas, “Modern family: A revocable hybrid encryption scheme based on attribute based encryption, symmetric searchable encryption and SGX,” in Security and Privacy in Communication Networks, S. Chen, K.-K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, Eds. Cham, Switzerland: Springer, 2019, pp. 472–486.
- [2]. J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proc. IEEE Symp. Secur. Privacy (SP). Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334.
- [3]. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, “Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list,” in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur. Cham, Switzerland: Springer, Jul. 2018, pp. 516–534.
- [4]. R. Dowsley, A. Michalas, M. Nagel, and N. Paladi, “A survey on design and implementation of protected searchable data in the cloud,” Comput. Sci. Rev., vol. 26, pp. 17–30, Nov. 2017.
- [5]. R. Bost, B. Minaud, and O. Ohrimenko, “Forward and backward private searchable encryption from constrained cryptographic primitives,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017, pp. 1465–1482.
- [6]. S. Agrawal and M. Chase, “FAME: Fast attribute based message encryption,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017, pp. 665– 682..
- [7]. G. Amjad, S. Kamara, and T. Moataz, “Forward and backward private searchable encryption with SGX,” in Proc. 12th Eur. Workshop Syst. Secur. (EuroSec). New York, NY, USA: Association for Computing Machinery, 2019.

[8]. A. Bakas and A. Michalas, “Multi-client symmetric searchable encryption with forward privacy,” Cryptol. ePrint Arch., Tampere Univ., Tampere, Finland, Tech. Rep. 2019/813, 2019. [Online]. Available: <https://eprint.iacr.org/2019/813>

[9]. A. Bakas and A. Michalas, “Power range: Forward private multi-client symmetric searchable encryption with range queries support,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jul. 2020, pp. 1–7.

[10]. A. Boldyreva, V. Goyal, and V. Kumar, “Identitybased encryption with efficient revocation,” in Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2008, pp. 417–426

[11]. D. Boneh, X. Boyen, and E.-J. Goh, “Hierarchical identity based encryption with constant size ciphertext,” in Advances in Cryptology—EUROCRYPT, R. Cramer, Ed. Berlin, Germany: Springer, 2005, pp. 440–456

[12]. ringer, 2005, pp. 440–456. [9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. Int. Conf. Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2004, pp. 506–522

[13]. V. Boyko, “On the security properties of OAEP as an all-or-nothing transform,” in Advances Cryptology—CRYPTO, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 503–518,

[14]. R. Bost, “ σ ϕ ϕ : Forward secure searchable encryption,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1143–1154.

[15]. M. Etemad, A. Küpçü, C. Papamanthou, and D. Evans, “Efficient dynamic searchable encryption with forward privacy,” Proc. Privacy Enhancing Technol., vol. 2018, no. 1, pp. 5–20, Jan. 2018.

[16] B. Fisch, D. Vinayagamurthy, D. Boneh, and S. Gorbunov, “IRON: Functional encryption using Intel SGX,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Oct. 2017, pp. 765–782.

[17] S. Lee, M. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, “Inferring fine-grained control flow inside SGX enclaves with branch shadowing,” in Proc. 26th USENIX Secur. Symp., Victoria, BC, Canada, Aug. 2017, pp. 557–574.

[18] A. Michalas, “The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing,” in Proc. 34th ACM/SIGAPP Symp. Appl. Comput., New York, NY, USA, Apr. 2019, pp. 146–155.

[19] A. Michalas, “Text files from Gutenberg database,” Tampere Univ., Tampere, Finland, Tech. Rep., Aug. 2019. [Online]. Available: <https://zenodo.org/record/3360392#.X7fuas0zaUk>

CERTIFICATES

	 CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (UGC-AUTONOMOUS) PRODDATUR Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360	 ICIAET-24 Proceedings
<p align="center">International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24) 5th & 6th April 2024</p> <p align="center">Organized by Department of Electrical & Electronics Engineering</p> <p align="center">Certificate of Appreciation</p>		
<p>This certificate is awarded to Dr./Mr./Mrs./Miss. <u>Narasimhulu Malavathula</u>, of <u>SRIT, Aranta pur</u> has participated and presented a paper entitled <u>Safe sharing : Access control for cloud stored data</u> with Paper ID: <u>ICIAET- P168</u> in ICIAET-2024.</p>		
 Dr V Mahesh Kumar Reddy Convenor & Organising Chair	 Dr G Sreenivasula Reddy Principal, CBIT	 Official sponsor



CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC-AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.),
Andhra Pradesh 516360



ICIAET-24
Proceedings

**International Conference on Innovative Approaches in
Engineering & Technology (ICIAET-24)**
5th & 6th April 2024
Organized by Department of Electrical & Electronics Engineering

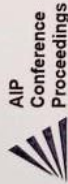
Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss Ajay Kishore Gattu of
SRI. Anantapur has participated and presented a paper entitled
Safe sharing: Access control for cloud stored data
_____ with Paper ID: ICIAET- P168 in ICIAET-2024.


Dr V Mahesh Kumar Reddy
Convener & Organising Chair


Dr G Sreenivasula Reddy
Principal, CBIT

 **turnitin**
Official sponsor



CHAITYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC-AUTONOMOUS)

PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.),
Andhra Pradesh 516360



ICIET-24
Proceedings

**International Conference on Innovative Approaches in
Engineering & Technology (ICIET-24)**
5th & 6th April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

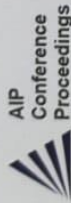
This certificate is awarded to Dr./Mr./Mrs./Miss Mounika Meelega, of
SRI. Ananta puri has participated and presented a paper entitled
Safe Sharing: Access control for cloud stored data
with Paper ID: ICIET-PI68 in ICIET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT



Official sponsor



CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC - AUTONOMOUS)
PRODDATUR

Vidya Nagar, Proddatur, YSR Kadapa (Dist.),
Andhra Pradesh 516360



ICAET-24
Proceedings

**International Conference on Innovative Approaches in
Engineering & Technology (ICAET-24)**
5th & 6th April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Hansha Sni Talanki of
SRI. Ananta pur has participated and presented a paper entitled
Safe Sharing: Access control for cloud stored data.
_____ with Paper ID: ICAET- P168 in **ICAET-2024**.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT



Official sponsor



CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.),
Andhra Pradesh 516360



ICIET-24
Proceedings

**International Conference on Innovative Approaches in
Engineering & Technology (ICIET-24)**

5th & 6th April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Bhavana Bommireni, of
SRIT, Anantapur has participated and presented a paper entitled
Safe Sharing : Access control for Cloud Stored Data.
_____ with Paper ID: ICIET-168 in **ICIET-2024**.


Dr V Mahesh Kumar Reddy
Convenor & Organising Chair


Dr G Sreenivasula Reddy
Principal, CBIT

 **turnitin**
Official sponsor