

# **SECURE FILE STORAGE ON CLOUD USING HYBRID CRYPTOGRAPHY**

**A PROJECT REPORT**

*Submitted by*

**PARTHIBAN.N (612819104027)**

**VEDIYAPPAN.C (612819104056)**

**VETRI.K (612819104057)**

**VIJAYSANKAR.G (612819104059)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**VARUVAN VADIVELAN INSTITUTE OF TECHNOLOGY**

**DHARMAPURI– 636703**

**ANNA UNIVERSITY:CHENNAI 600025**

**MAY 2023**

# **ANNA UNIVERSITY:CHENNAI 600025**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**SECURE FILE STORAGE ON CLOUD USING HYBRID CRYPTOGRAPHY**” is a bonafide work of “**PARTHIBAN N (612819104027), VEDIYAPPAN C (612819104056), VETRI K (612819104057), VIJAYSANKAR G (612819104059)**” who carried out the project work under my supervision.

### **SIGNATURE**

**Mrs.M.GEETHARANI, M.E.,  
HEAD OF THE DEPARTMENT**

Department of Computer  
Science & Engineering  
Varuvan Vadivelan Institute of  
Technology,Dharmapuri

### **SIGNATURE**

**Mr.L.D.RAJA, M.E.,  
SUPERVISOR**

**Assistant Professor**  
Department of Computer  
Science & Engineering  
Varuvan Vadivelan Institute of  
Technology,Dharmapuri

Submitted for Anna University project Viva-voce held on.....  
.....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We would like to acknowledge the constant support provided by our Chairman **Thiru.M.VADIVELAN, M.B.A.**, without whom we could not have completed this project in time.

We feel elated to keep on record our heartfelt thanks and gratitude to our principal **Dr.S.RAGUNATHAN, M.E., Ph.D.**, who has been our constant source of inspiration.

We would like to thank our director **Dr.A.SIVAKUMAR, M.E., Ph.D.**, and vice principal for providing the necessary facilities whenever needed to carry out our project.

We express our regards to **Mrs.M.GEETHARANI, M.E.**, Head of the Department in Computer Science and Engineering for providing the necessary facilities whenever needed to carry out the project.

We are also grateful to have **Mrs.P.KEERTHIKA, M.E.**, Asst.Professor in Department of Computer Science & Engineering as my project coordinator. Her appreciation right from the beginning has been the source of energy.

We would like to thank our guide **Mr. L.D.RAJA, M.E.**, Asst.Professor in Department of Computer Science & Engineering for his valuable guidance and encouragement during the entire span of this project.

## **ABSTRACT**

## **ABSTRACT**

Currently, data security and privacy policy has been regarded as one of the biggest concerns in cloud computing. Data stored at remote storage is unsafe and susceptible to get hacked. Due to this, users do not trust their data over the cloud. Cloud consumers want an assurance that they can access their data wherever they want and no one else is able to get it. After doing the survey and studying the research papers it is found that the major security concerns of cloud computing include Data leakage, Distributed Denial of Service (DDoS). A hybrid model is proposed which is a mixture of elliptical curve cryptography and symmetric key algorithm. ECC is used to achieve the process of user's verification and to keep the private data secure. AES algorithm is used which allows the user to store and access their data securely to the cloud by encrypting the data in the client side and decrypting the data after downloading from the cloud. Since the private key is owned by the user of the data, no one can decrypt the data, even though the hacker can get the data through some approaches. Moreover, user will securely authenticate itself. This scheme can make users assure about the security of data stored in the cloud. Here, we will apply an ECC and ECDH algorithm that provide same level of security as of other public key crypto systems with less key size and strengthens.

## LIST OF FIGURES

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
5.1	Overall display of the project	24
5.2	Overall Architecture	24
5.3	Use-Case Diagram	25
5.4	Auto ID Increment	25
5.5	Users Files	26
5.6	Users Table	26
5.7	User Login	27
6.1	Home page	28
6.2	Registration page	28
6.3	Key exchange	29
6.4	ECDH key aggrement	29
6.5	User ID	30
6.6	Login page	30
6.7	User Account	31
6.8	File Encryption	31
6.9	File Decryption	32
6.10	File Download	32
7.1	Program Execution	38
7.2	Home Page	38

## **LIST OF ABBREVIATIONS**

<b>ABBREVIATIONS</b>	<b>EXPANSION</b>
AES	Advance encryption standards
DES	Data encryption standards
DDOS	Distribute denial of services
ECC	Elliptic curve cryptography
ECDH	Elliptic-curve Defi-Hellman
SAAS	Software as a service
PAAS	Platform as a service
IAAS	Infrastructure as a service
CSP	Cloud service provider
NIST	National Institute of Standards and Technology

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>I</b>
	<b>LIST OF FIGURES</b>	<b>II</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>III</b>
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 PURPOSE	1
	1.2 OBJECTIVE	3
	1.3 MOTIVATION	4
	1.4 DEFINITION AND OVERVIEW	5
<b>2</b>	<b>RELATED WORK</b>	
	2.1 MAJOR SECURITY CONCERNS OF CLOUD COMPUTING	7
	2.2 DATA PRIVACY IN CLOUD	9
	2.3 LITRATURE SURVEY	10
<b>3</b>	<b>SYSTEM ANALYSIS</b>	
	3.1 EXISTING SYSTEM	12
	3.2 PROPOSED SYSTEM	13
<b>4</b>	<b>PROJECT SPECIFICATION</b>	
	4.1 EXTERNAL INTERFACE	14
	4.2 FUNCTIONAL REQUIREMENTS	15
	4.3 NON – FUNCTIONAL REQUIREMENTS	16



5	<b>PROBLEM DESIGN</b>	
	5.1 ALGORITHM / METHODOLOGY	18
	5.2 PROJECT FUNCTIONS	22
	5.3 CLASS TABLES AND UML DIAGRAMS	24
	5.4 CONSTRAINTS AND ASSUMPTIONS	27
6	<b>RESULT AND DISCUSSIONS</b>	
	6.1 AUTHENTIC REGISTRATION OF THE USERS	28
	6.2 PREVENTING DDOS ATTACK USING SECURE LOGIN VIA OTP	29
	6.3 FILE UPLOADING AND DOWNLOADING WITH AES	31
7	<b>APPENDICES</b>	33
8	<b>CONCLUSION AND FUTURE WORK</b>	
	8.1 CONCLUSIONS	39
	8.2 SCOPE FOR FUTURE WORK	40
	<b>REFERENCES</b>	41

# **CHAPTER 1**

## **INTRODUCTION**

# **CHAPTER 1**

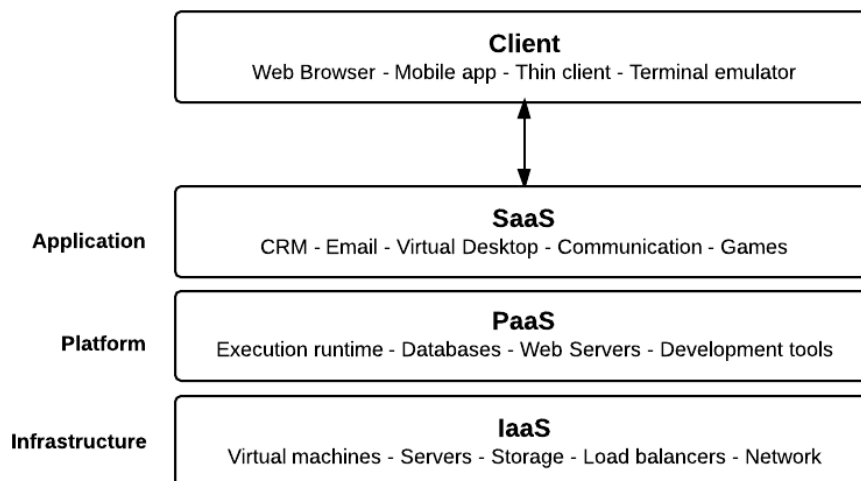
## **INTRODUCTION**

Cloud Computing is the style of computing where the resources are provided as services on internet. There are three types of services in Cloud Computing which are used for the deployment of the application on the cloud. Data on the cloud will become more scalable, Reliable and Secure. The big players in Cloud Computing are Amazon, Google, Microsoft and IBM. Cloud Computing is based on five attributes such as Shared Resources, Scalability, Pay as U use, Elasticity and Self Provisioning of Resource. Most of the enterprises shifting their applications on to the cloud owing to its speed of implementation and deployment, improved customer experience, scalability, and cost control. The services in Cloud Computing are SaaS, PaaS, IaaS amongst which we are using PaaS and IaaS service for deployment of Application on the Cloud in our Project. This service exhibits five essential characteristics such As Rapid Elasticity, Resource Pooling, on demand Self-service, Broad Network Areas. Data is being transmitted between two clouds so in order to secure the data most of the systems use the combination of techniques, including: Encryption is used to encode the data in such a way that third party will not be able to hack that data. Authentication is used to create a separate user ID and Password so that only the authorized users will able to access the data. Separation of duties which accessibility is provided to all the users according to their priority. These security parameters are achieved due to which the performance will gets increased and therefore the Security is obtained up to higher extent. Data security and privacy risks have become the primary concern for people to shift to cloud computing. Cloud Computing is mainly used for the improving the data handling capability where the services and the resources will be delivered continuously when and where required due to which the Cloud computing is in great demand.

However there still exist many problems in cloud computing today, a recent survey 3 shows that data security and privacy risks have become the primary concern for people to shift to cloud computing Cloud is the free space where the application is being saved securely and the services are being provided continuously when and where required.

### Cloud Deployment Models:

- **Public Cloud** - The Cloud infrastructure is made available for the large industry group and general public provided by single service provider.
- **Private Cloud** - The Organization can store the data on private Cloud. The main Advantage of this Cloud is Security of Data and Quality of Service.
- **Community Cloud** - The Cloud Infrastructure is shared by many Organizations.
- **Hybrid Cloud** - Two or more Clouds combine to form Hybrid Cloud.



### Cloud platform services

## Cloud Characteristics:

- **Easy Use**- Most Cloud Provider will offer the Internet interfaces which are much simple so user can easily access the cloud services.
- **Ubiquitous Network Access** - Cloud provides services through the standard terminal such as phones, Laptops, Mobiles.
- **On demand Services** - Cloud is a pool of resources and services so we can get the services and resources by paying particular amount as required.
- **Business Model** - Cloud is a Business Model because it is pay per use of service or resource.
- **Pay as U Used** - Users have to pay for only the Resources they are using. Whenever the users need some resources then they have to pay for the particular resource as and when required.

## 1.2 OBJECTIVE

The primary goal of this project is to provide and simulate an effective solution to face the challenges and solve security issues that exists in cloud computing. Cloud Computing is the impending need of computing which is used for the IT Industries It is one of the hottest topic in research areas. Scalability and Flexibility increases for the computing services. Cloud Computing is the fastest growing technology for IT Industry. The Information is being transmitted via the network therefore security is one of the main problems or issue. The Application is deployed on the Cloud and for the secure transmission of the data we will be using ECC Algorithm in our project because of its advantages in terms of CPU utilization, time for Encryption and Key Size. This Project will explore the deployment of Application on the Cloud and increases the security level by implementing ECC & ECDH Algorithm, and AES Algorithm for secure file handling and Encryption.

### 1.3 MOTIVATION

Need of data security is an essential issue in the domain of computing traditionally. There are various algorithms are developed in order to improve the security of data, but they having their 5 own issues. Now in these days the traditional algorithms are not much suitable for providing security over the untrusted communications and data exchange. ECC is more and more considered as an attractive public-key cryptosystem for mobile/wireless environments. One of the other recent public key cryptosystems is Elliptic Curves Cryptography use for security. In recent times, the majority of e-commerce applications are designed using asymmetric cryptography to assure the authentication of the concerned parties. Compared to traditional public-key cryptosystems like RSA or Diffie-Hellman, ECC propose equivalent security with smaller key sizes; these results in faster calculation, lower power expenditure, as well as memory and bandwidth savings. ECC is peculiarly useful for mobile devices, which are typically particular in terms of their CPU, power and network connectivity. Therefore, a new encryption standard is required that can fulfil the current need of security meanwhile that is extendable according to the need. The proposed work includes the development of new hybrid algorithm using ECC, ECDH and AES algorithms along with encryption techniques.

## **1.4 DEFINITION AND OVERVIEW**

Cloud Computing is the primitive change happening in the field of Information Technology. It uses the internet technologies for delivery of IT - enabled capabilities 'as a service' to any needed users. Cloud computing enables users to access resources using internet, from anywhere at any time without worrying about technical/physical management and maintenance concern of the original resources. In its description for cloud characteristics The US National Institute of Standards and Technology (NIST) defines as cloud characteristics the following: On-demand self-service, Ubiquitous network access, Resource pooling, Rapid elasticity (resources can be scaled up and down easily), Metered service (resources' usage is measured) and Pay-As-YouConsume business models. Google Apps is an important example of Cloud computing; it enables to access services through the browser and brought into effective action on millions of machines over the Internet. One of the most prominent service offered by cloud computing is cloud 6 storage. Cloud storage is simply a term that refers to on line space that you can use to store your data. In more strict way, cloud storage is a service model in which data is maintained, managed and backed up remotely and made available to users over a network.

### **Software as a service (SaaS)**

The cloud provider deploys, configures, maintains and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The cloud consumers have limited administrative control of the applications.

### **Platform as a service (PaaS)**

The Cloud Provider manages the computing infrastructure for the platform and runs the cloud software that provides the components of the platform, such as runtime software execution stack and databases.

The PaaS Cloud Provider typically also supports the development, deployment and management process of the PaaS Cloud Consumer by providing tools such as integrated development environments (IDEs), development version of cloud software, software development kits (SDKs), deployment and management tools.

### **Infrastructure as a service (IaaS)**

The Cloud Provider acquires the physical computing resources underlying the service, including the servers, networks, storage and hosting infrastructure. The Cloud Provider runs the cloud software necessary to make computing resources available to the IaaS Cloud Consumer through a set of service interfaces and computing resource abstractions, such as virtual machines and virtual network interfaces. The whole idea and definition of this project lies in its name i.e. Secure Cloud, which aims at providing and simulating an effective solution to face the challenges and solve security issues that exist in cloud computing. But first we should look at some of the frequently occurring issues in cloud computing mostly during the transmission of data. Some of them are discussed below:

- **Encryption** - The message sent by the sender i.e. the original message is being encrypted in such a way that third party will not be able to hack or misuse the data.
- **Intrusion Detection and Prevention** - Data that is being entered and going out of the Network has to be known.
- **Separation of Duties** - Due to the insufficient communication between the expertise System misconfiguration takes place.
- **Location of Data** - Every Organization will have different requirements and their access control on their data to be placed. A level of security is required to fulfil the customer need.



## **CHAPTER 2**

### **RELATED WORK**

## **CHAPTER 2**

### **RELATED WORK**

This chapter surveys previous work done in the field of major security concerns in cloud computing and how to ensure data privacy using various cryptographic algorithms.

#### **2.1 MAJOR SECURITY CONCERNS OF CLOUD COMPUTING**

##### **Data Leakage**

Innately, when moving to a cloud there is two changes for customer's data. First, the data will store away from the customer's local machine. Second, the data is moving from a single-tenant to a multi-tenant environment. These changes can raise an important concern that called data leakage. Because of them, Data leakage has become one of the greatest organizational risks from security standpoint.

##### **Cloud Security Issues**

Innately, Internet is communication infrastructure for cloud providers that use well known TCP/IP protocol which users' IP addresses to identify them in the Internet. Similar to physical computer in the Internet that have IP address, a virtual machine in the Internet has an IP address as well. A malicious user, whether internal or external, like a legal user can find this IP addresses as well. In this case, malicious user can find out which physical servers the victim is using then by implanting a malicious virtual machine at that location to launch an attack. Because all of users who use same virtual machine as infrastructure, if a hacker steals a virtual machine or take control over it, he will be able to access to all users' data within it. Therefore, The hacker can copy them into his local machine before cloud provider detect that virtual machine is in out of control then the hacker with analysis the data may be find valuable data afterward.

## **Attacks in Cloud**

Nowadays, there are several attacks in the IT world. Basically, as the cloud can give service to legal users it can also service to users that have malicious purposes. A hacker can use a cloud to host a malicious application for achieve his object which may be a DDoS attacks against cloud itself or arranging another user in the cloud. For example, assume an attacker knew that his victim is using typical cloud provider, now attacker by using same cloud provider can sketch an attack against his victim. This situation is similar to this scenario that both attacker and victim are in same network but with this difference that they use virtual machines instead of physical network.

### **DDoS Attacks against Clouds**

Distributed Denial of Service (DDoS) attacks typically focus high quantity of IP packets at specific network entry elements; usually any form of hardware that operates on a Blacklist pattern is quickly overrun and will become in out of - service situation. In cloud computing where infrastructure is shared by large number of clients, DDoS attacks make have the potential of having much greater impact than against single tenanted architectures

### **Cloud against DDoS Attacks**

DDoS attacks are one of the powerful threats available in world, especially when launched from a botnet with huge numbers of zombie machines. When a DDoS attack is launched, it sends a heavy flood of packets to a Web server from multiple sources. In this situation, the cloud may be part of the solution. it's interesting to consider that websites experiencing DDoS attacks which have limitation in server resources, can take advantage of using cloud that provides 24/7 more resource to tolerate such attacks. In the other hand, cloud technology offers the benefit of flexibility, with the ability to provide resources almost instantaneously as necessary to avoid site shutdown.

## 2.2 DATA PRIVACY IN CLOUD

Sharing of Cloud Infrastructure could lead to the privacy issues. The Location of data could influence the privacy obligations. For storage and processing of data. Data leakage could also occur due to failure of security access rights[5]. In order to secure the data stored on the cloud various security Algorithms are present which will help to encrypt the data before transmission in order to protect the valuable data from the hackers. One of the better solution for maintain the security is cryptography which is basically used for protecting the data.

- **Public Key Cryptography** - In this cryptography different keys are used for Encryption and Decryption.
- **Secret key cryptography** - A key which is used for encryption as well as decryption is called secret key cryptography.

There are many Security Algorithms Each Algorithm have their own properties such as Key Size, Throughput, Performance, Encryption Decryption Time etc. By Comparing the Encryption Algorithms we found out that ECC Algorithm is one of the best Algorithm which is having the high level of Security and better performance.

After doing the survey and studying the research papers it is found that the major security concerns of cloud computing includes Data leakage, Distributed Denial of Service (DDOS). The data security can be improved by implementing various symmetric key algorithms so that data on the server is stored in a manner that even if a person gets access then also he can't open the original data. As it needs to be decrypted.

## **2.3 LITRATURE SURVEY**

Cloud computing has become a popular trend for data storage and sharing. However, ensuring the security and confidentiality of data stored on the cloud remains a major concern. Hybrid cryptography, which combines symmetric and asymmetric cryptography, has been proposed as a potential solution to address this issue. In this literature survey, we will explore the various research studies that have been conducted on secure file storage on the cloud using hybrid cryptography.

### **"Hybrid Cryptography Based Cloud Storage Security Mechanism for Privacy and Integrity" by S. Ramar and S. K. Srivatsa**

This paper proposes a hybrid cryptography-based security mechanism for cloud storage. The proposed mechanism uses AES and RSA algorithms to ensure the confidentiality and integrity of data stored on the cloud. The authors implemented the proposed mechanism and evaluated its performance using various metrics.

### **"A Hybrid Cryptographic Technique for Secure Cloud Storage" by S. Z. Ismail, A. G. Radwan, and A. A. Ewees**

This paper proposes a hybrid cryptographic technique for secure cloud storage that combines symmetric-key and public-key cryptography. The proposed technique uses the AES and RSA algorithms for encryption and decryption of data stored on the cloud.

**"Enhanced Security of Cloud Storage using Hybrid Cryptography" by A. A. Alhammadi and S. B. Gaid**

This paper proposes an enhanced security mechanism for cloud storage using hybrid cryptography. The proposed mechanism uses the AES and RSA algorithms for encryption and decryption, and the HMAC-SHA256 algorithm for message authentication.

**"Secure File Storage on Cloud using Hybrid Cryptography" by R. R. Giri and A. B. Gudadhe**

This paper proposes a secure file storage mechanism on the cloud using hybrid cryptography. The proposed mechanism uses the AES and RSA algorithms for encryption and decryption, and the SHA-512 algorithm for message authentication. The authors evaluated the proposed mechanism and compared it with other encryption techniques.

**"A Comparative Study of Encryption Algorithms for Secure Cloud Storage" by S. Arora and A. K. Sharma**

This paper presents a comparative study of various encryption algorithms, including hybrid cryptography, for secure cloud storage. The authors evaluated the performance of different encryption algorithms using various metrics such as encryption and decryption time, throughput, and security. The study provides insights into the effectiveness of hybrid cryptography compared to other encryption.

# **CHAPTER 3**

## **SYSTEM ANALYSIS**

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

When a user uploads data, it is divided into three sections, the first of which is encrypted with AES, the second with DES, and the third with RSA. LSB steganography is used to store the keys in the image, and the three encrypted— files are stored in the cloud. Users must first recover the keys from the image before they can import all data from the server. These keys are then used to decrypt the data once more with AES, DES, and RSA.— This approach increases the security of records.

#### **DISADVANTAGES**

- User selects the file from the local storage.
- The file will be uploaded to the cloud after getting encrypted.
- It uses RSA algorithms.
- It using steganography to hide the key.
- It needs some more time for encryption and decryption process.



### 3.2 PROPOSED SYSTEM

In proposed system, Here, we will apply an ECC and ECDH algorithm that provide same level of security as of other public key crypto systems with less key size and strengthens the security of the algorithm. The whole prototype of the proposed solution would benefit by enabling a proper access mechanism to avoid unauthorized access to the information system and a secure storage to allow access of data over the cloud network.

- **Encryption** - It is used to encode the data in such a way that third party will not be able to hack that data.
- **Authentication** - It is used to create a separate user ID and Password so that only the authorized users will able to access the data.

### ADVANTAGES

- The proposed system is designed to provide high security to the data.
- It converts the plain text into a cipher text and key store it into the cloud.
- Implementing ECC & ECDH Algorithm, and AES Algorithm for secure file handling and Encryption.
- The secure transmission of the data we will be using ECC Algorithm.
- Its advantages in terms of CPU utilization, time for Encryption and Key–Size.

# **CHAPTER 4**

## **PROJECT SPECIFICATION**

## **CHAPTER 4**

### **PROJECT SPECIFICATION**

#### **4.1 EXTERNAL INTERFACE REQUIREMENT**

##### **4.1.1 User Interface**

- Platform – Desktop or Mobile browser.
- Display – 1024x768 or higher
- JavaScript enabled browser.
- Latest Video and supported graphics drivers.

##### **4.1.2 Hardware Interface.**

- Processor – i3/i5/i7 x64 Bit Minimum 2 Ghz.
- Hard Disk – 256 GB + at least 2 GB for Relational Database.
- Memory – 4 GB RAM Recommended
- High Speed Internet Access.
- LAN Connection with Ethernet.

##### **4.1.3 Software Interface**

- Linux / Windows OS, Ubuntu 16.04 Preferred.
- JDK 7 or above.
- NetBeans IDE.
- AWS (Relational Database Server), MYSQL Preferred.
- Apache Tomcat Server.

## **4.2 FUNCTIONAL REQUIREMENTS**

The functional requirement part discusses the functional behavior that should be possessed by the system. Each requirement maps to a higher-level function that transforms the given set of input data into output data. The functional requirements can be identified as the modules involved.

### **Module 1: Introduction Module**

- Purpose – A brief introduction. It is invented to be engaging and communicate the theme of the cloud application to the user.
- Inputs – No input is necessary.
- Outputs – Immediately load the Main Menu Screen (Registration Screen).

### **Module 2: Registration Module**

- Outputs – Control is passed to key exchange page with a random registration created.
- Inputs – Username, Mobile Number, Email, DOB fields are displayed, submit button.

### **Module 3: Key Exchange Module**

- Purpose – For ECCDH equivalent key exchange.
- Inputs – Secret Private Key for exchange.
- Outputs –ECDH Key is generated, and OTP sent to mail ID.

### **Module 4: ID Generation Module**

- Purpose – For user ID generation. Generation of user ID. Accessing the cloud storage. Fresh OTP sent to email ID.
- Inputs – OTP from email ID in the text field. User ID and OTP
- Outputs – Random user ID is generated. OTP verification and redirecting to user account.

### **Module 5: Login Module**

- Purpose – To check credentials of the user and log him in if they are correct and grant the access to their account.
- Inputs – User ID and the OTP sent to the user's email ID.
- Outputs – Immediately load the Profile Screen if the credentials match.

## **4.3 NON - FUNCTIONAL REQUIREMENTS**

A careful specification and adherence of non-functional requirements such as performance, security, privacy and availability are crucial to the success or failure of any software system.

### **4.3.1 Performance Requirements**

- The capability of the application depends on the performance of the servers. Anyone can use the application easily because of good GUI.
- The application can take any number of users provided the database size is large enough. It depends on the available memory space in database.
- On mobile devices and laptops, the battery is a scarce and valuable resource. The battery should remain maximally available for the application to perform well. Your application may therefore fall by the wayside or even get uninstalled by the user, if it drains too much battery.
- The text font size may need to be adjusted up (for high resolution screens) or down (for low resolution screens) to keep the text readable.

### **4.3.2 Safety Requirements**

- The layout may need to be taken care of and adjusted to increase or decrease the spacing between and around labels and widgets shown on the screen so as to prevent them from getting clustered together on high-res screens or spaced apart too much on low-res screens.
- Any images used in the project have to be provided in two different versions: a large size/high resolution version and a small size/low resolution version so that it properly fills the amount of physical space available on the screen.

### **4.3.3 Security Requirements**

- Although security is the utmost priority and has been taken care the most but care must be taken against virus and malware threats.
- This application will be available for all the users within the Internet. The system server should be up for 365 days and the downtime should be minimized in case of any attack or difficulties.
- Firewall should be used on the user's system to prevent any suspicious activity.

#### **4.3.4 Software Quality Attributes**

- 24x7 availability of the system with suitable updating at regular interval of time. To maintain integrity of the data and in order to ensure the security of the database by asking them to sign up for the application.
- Form validation so that only real users access the system. An error message should be displayed in case of improper working of the application.
- Email -ID entered should be valid as OTP is sent to that Email ID.
- The application can be accessed at any place that has Internet connectivity. Always save the data before closing the website.
- An error message should be displayed in case of improper working of the application.
- 24 hours availability of internet connection is required.

# **CHAPTER 5**

## **PROBLEM DESIGN**

## **CHAPTER 5**

### **PROBLEM DESIGN**

#### **5.1 ALGORITHM/METHODOLOGY**

We have in mind a hybrid model proposed which is a mixture of Elliptical Curve Cryptography and a symmetric key algorithm. ECC is used to achieve the process of user's verification and to keep the private data secure. AES algorithm is used which allows the user to store and access their data securely in the cloud by encrypting the data in the client side and decrypting the data after downloading from the cloud.

Since the private key is owned by the user of the data, no one can decrypt the data. Even though hackers can get the data through some approaches, they will not be able to access it. Moreover, user will securely authenticate itself by using different input parameters at the time of login to the cloud server. This scheme can make users assure about the security of data stored in the cloud. Here, we will apply an ECC and ECDH algorithm that provides same level of security as of other public key cryptosystems with less key size and strengthens the security of the algorithm.

Benefits are:

- Proper access mechanism to avoid unauthorised access to the information system
- Secure storage and access of data

The model would be hosted on a website. The website is designed keeping in mind an average user so that he can understand the functioning easily and does not have to hassle with the complex design of the website. The simple design of the website makes it very easy for any user to access it easily and make full use of the product.



## **Elliptic Curve Cryptography (ECC):**

Elliptic curve systems were first proposed in 1985 by Neal Koblitz and Victor Miller. An Elliptic curve over a field  $K$  have a set of points  $(X_i, Y_i)$  in a plane. The set is finite and is denoted by  $E$ . It is one of the most secure Algorithm. ECC is a public key cryptography Algorithm in which each and every user has its own pair of private and public key. Group Operator is an important one in ECC and is denoted by the symbol '+'. The Standard form of ECC is given by  $y^2 = x^3 + ax + b$  for some fixed values for parameters  $a$  and  $b$ . The security of ECC Algorithm depends on the ability of computation of new points on the curve and then the encryption of these points as information is to be exchanged between the end users. Group Operator is used to find  $P$  which is one of the point on the curve. Again, this operator proceeds the computation as  $P+P, P+P+P, \dots$ , which makes it very difficult for the hacker to hack the data.

## **Key Agreement using ECDH Algorithm:**

Both clouds i.e. Cloud A and Cloud B will agree for the data which is being transmitted The Agreement between the two parties will takes place only when both the keys are same. 1) A will select an integer  $X_A = k_1$  as his/her private key. The public key for A will be  $Y_A = X_A \times P$ , which implies that when the private key is an ordinary integer, the public key is a point like  $P$ . 2) B does exactly the same thing it selects an integer  $X_B = K_2$  as his/her private key, with the public key for B being  $Y_B = X_B \times P$ . Then both the parties exchange their public keys. 3) A computes the session key by  $K_A = X_A \times Y_B = k_1 \times k_2 \times P$  4) B computes the session key by  $K_B = X_B \times Y_A = k_2 \times k_1 \times P$ . Obviously,  $K_A = K_B$  This proves the Agreement for exchanging the Data between two parties and the generation of public and private key.

## **Key Generation:**

Key generation is an important part where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key and the receiver will decrypt its private key. Now, we have to select a number 'd' within the range of 'n'. Using the following equation we can generate the public key.

$$Q = d * P$$

d = The random number that we have selected within the range of ( 1 to n-1 ). P is the point on the curve.

'Q' is the public key and 'd' is the private key.

Algorithm generates both the public key and private key. Here Sender will be used to encrypt the data and receiver i.e. B is used to decrypt the data by using its own private key.

## **Encryption:**

Let 'm' be the message that has been sent from the sender A to B. Sender A will encode the message and on the way of transmission only the encryption will take place and for the transmission of data only few nano-seconds will be required to travel the data to receiver.

## **File Encryption using AES**

AES is the Advanced Encryption Standard, a United States government standard algorithm for changing the plain text to cipher text i.e. encrypting and decryption the data. The proposed system uses AES to encrypt the files with a secret key to secure the uploaded files. The user has to enter the private key and select the file for encryption. Once the file is encrypted, during the process of decryption, the user has to enter the AES key which was generated during encryption.

## ALGORITHM

Algorithm Start (FileName, Private Key){

    Browse Files

    Enter private key

}

Algorithm Encrypt() {

    Convert to State Array

    AddRoundKey()

    SubBytes()

    ShiftRows()

    MixColumns()

    Key Expansion

}

Algorithm Decrypt {

    Convert to State Array

    AddRoundKey()

    InvSubBytes()

    InvShiftRows()

    InvMixColumns()

    Key Expansion

}

## 5.2 PROJECT FUNCTIONS

The landing page of the web application – the very first page which is visible when the app is launched in the browser of the user's system. This page lets the user know about the application and let the user to choose between first time registration and login features. Along with this, the page contains tabs for download and upload section also. It contains, app logo, tag line and makes the user aware about the goal of the application is short period of time and enhances aesthetic value.

The Registration Page of the application which is displayed when user is using the application for the first time or does not possess a user account. The form here takes user's full name, email address, mobile number, DOB and gender as input to store user's information and validate the user before giving access to the app data. It first checks for all possible errors in the credentials on the client side itself using regular expressions and pattern matching. Later, if the data passes all the test cases, it is sent to server side to validate and store the details in the database to create a working profile of the user.

The Registration Page of the application which is displayed when user is using the application for the first time or does not possess a user account. The form here takes user's full name, email address, mobile number, DOB and gender as input to store user's information and validate the user before giving access to the app data. It first checks for all possible errors in the credentials on the client side itself using regular expressions and pattern matching. Later, if the data passes all the test cases, it is sent to server side to validate and store the details in the database to create a working profile of the user.

The Registration Page of the application which is displayed when user is using the application for the first time or does not possess a user account. The form here takes user's full name, email address, mobile number, DOB and gender as input to store user's information and validate the user.

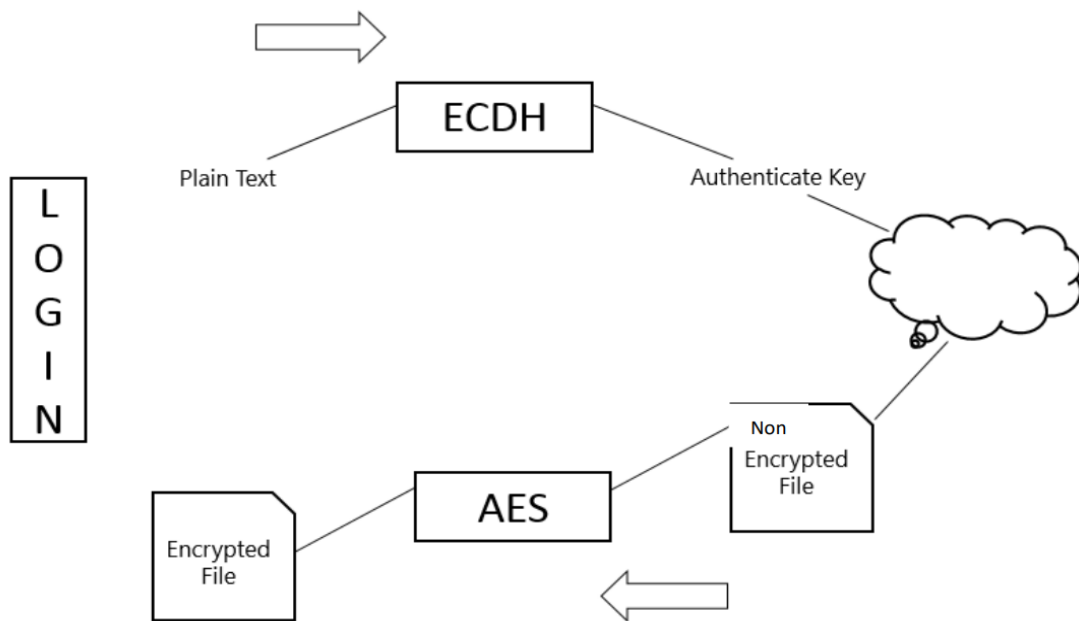
It first checks for all possible errors in the credentials on the client side itself using regular expressions and pattern matching. Later, if the data passes all the test cases, it is sent to server side to validate and store the details in the database to create a working profile of the user.

On successful validation, the application generates a unique User ID. The user is requested to save this ID as, once the registration is done, the user will be using just the User ID generated here and the secret key entered earlier to use all the features and perform validation further in the application. The new user registration completes here and now the user can easily login to access his profile and use all the features of the application.

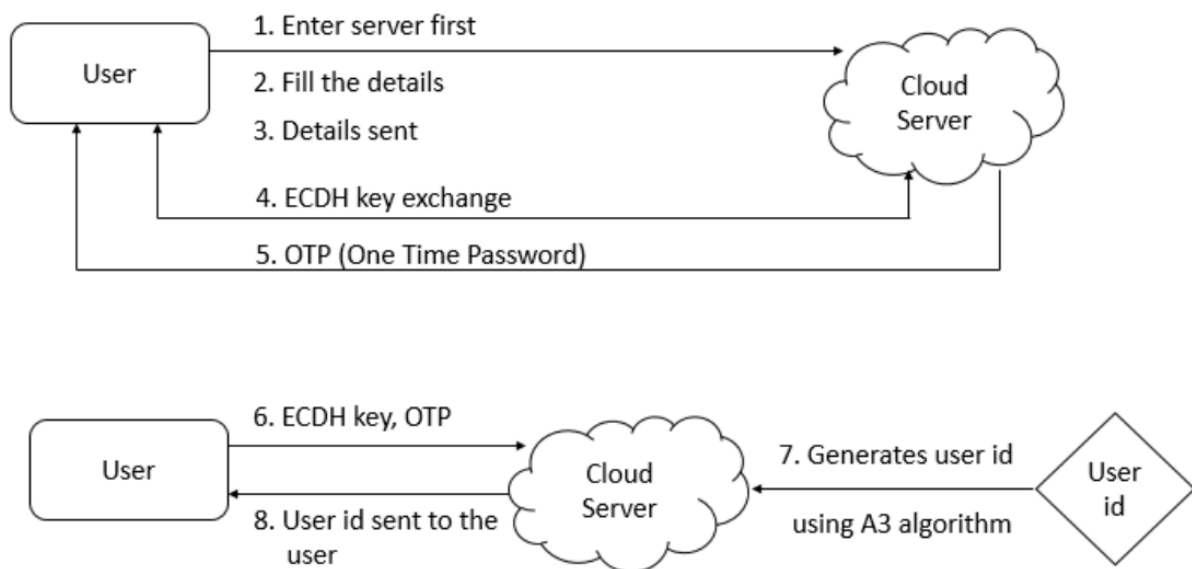
After that, the Login page of the application which is loaded when the user clicks on the login tab button. The page consists of a simple form which inputs the User ID of the user generated during registration process and the OTP which is immediately sent to the user's email address as soon as he clicks on 'Request for OTP' button in the form. Later, both the values in the fields are validated over the server with the values in the database and if successful the user is taken to the next page that is Dashboard of the application, else error is generated.

The Dashboard Page of the application. When the user is able to successfully login into the app for the first time, he is taken to the My Account section where he is shown the current details of his account stored in the database and is given an opportunity to update any field if required and click on submit button to save. The dashboard page serves as a navigating page as from here the user can go-to encryption, decryption download and upload sections and can log out from the app if the work is done.

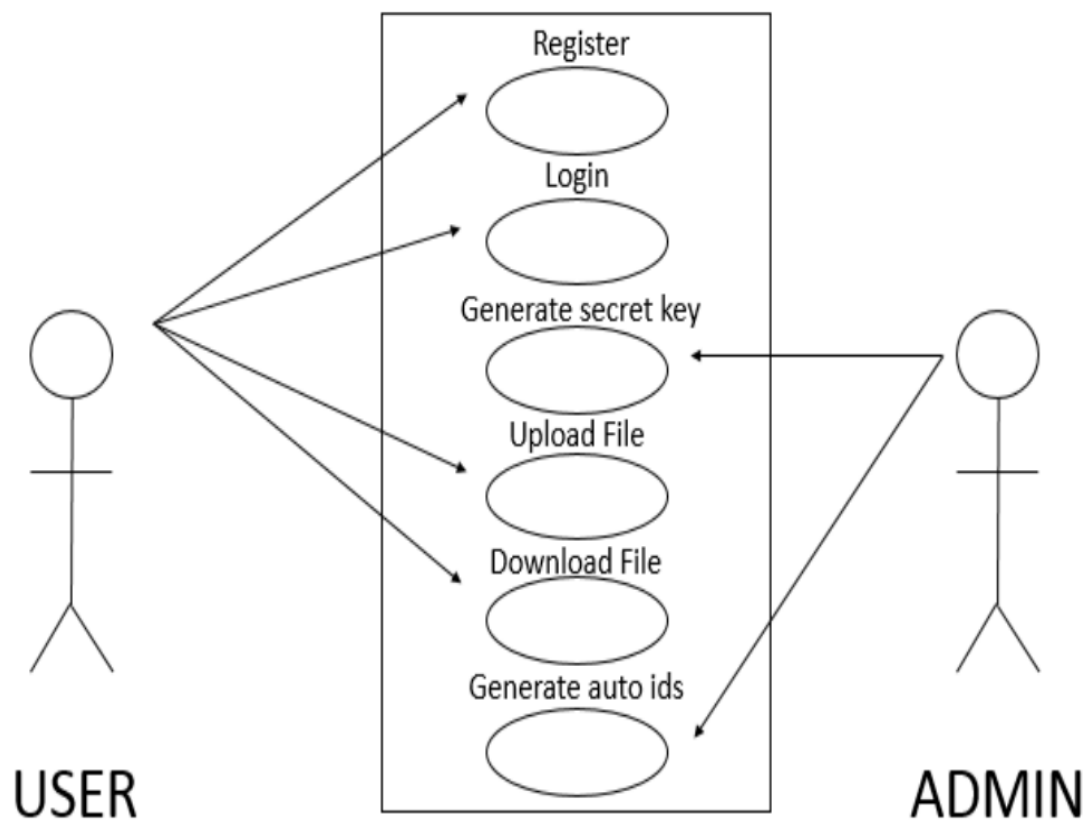
### 5.3 CLASS TABLES AND UML DIAGRAMS



**Fig 5.1 Overall display of the project**



**Fig 5.2 Overall architecture with component description and dependency details**



**Fig 5.3 User-Case Diagram**

### 5.3.2. CLASS TABLES

AUTO_ID		
NAME	DATA TYPE	NULL/ NOT NULL
FORM_NAME	Varchar(50)	Null
PREFIX	Varchar(100)	Null

**Fig 5.4 Auto ID increment**

FILES		
NAME	DATA TYPE	NULL/ NOT NULL
ID	Bigint(20)	Not Null
USER_ID	Varchar(50)	Null
FILE_NAME	Varchar(100)	Null
FILE_PATH	Varchar(255)	Null
CREATED	Datetime	Null
F_STATUS	Char(1)	1
ENCRYPTED_AES_KEY	Varchar(255)	Null
PUBLIC_KEY	Varchar(255)	Null
PRIVATE_KEY	Varchar(255)	Null

**Fig 5.5 Users File**

USERS		
NAME	DATA TYPE	NULL/ NOT NULL
USER_ID	Varchar(20)	Not Null
GEN_USER_ID	Varchar(200)	Null
NAME	Varchar(100)	Not Null
EMAIL	Varchar(100)	Null
MOBILE	Varchar(20)	Null
DOB	Varchar(100)	Null
GENDER	Varchar(50)	Null
USER_KEY	Varchar(50)	Null
SECRETU	Varchar(200)	Null
USER_OTP	Varchar(10)	Null
UTYPE	Varchar(20)	'user'
U_STATUS	Char(1)	1
CREATED	datetime	Null

**Fig 5.6 Users table**



USER_LOGIN		
NAME	DATA TYPE	NULL/ NOT NULL
USER_ID	Varchar(20)	Not Null
OTP	Varchar(10)	Not Null
CREATED	DATE	NOT NULL

**Fig 5.7 User login**

## 5.4 CONSTRAINTS AND ASSUMPTIONS

Following are the constraints of Secure File Storage Using Hybrid

Cryptography:

- The software is only available on the web.
- The project implements a security model and shows the simulation.
- To use this model in real time a lot of modifications will be needed.
- The project will be tested against a certain set of test cases only.

Following are the assumptions:

- The user has good knowledge of operating a computer and web application.
- The computer system has internet and/or LAN connection enabled.
- It is mandatory for the internet to be turned on and active.
- The user device has enough memory available for installation and proper functioning of the application.

# **CHAPTER 6**

## **RESULT AND DISCUSSIONS**

## CHAPTER 6

### RESULT AND DISCUSSIONS

#### 6.1 Home page

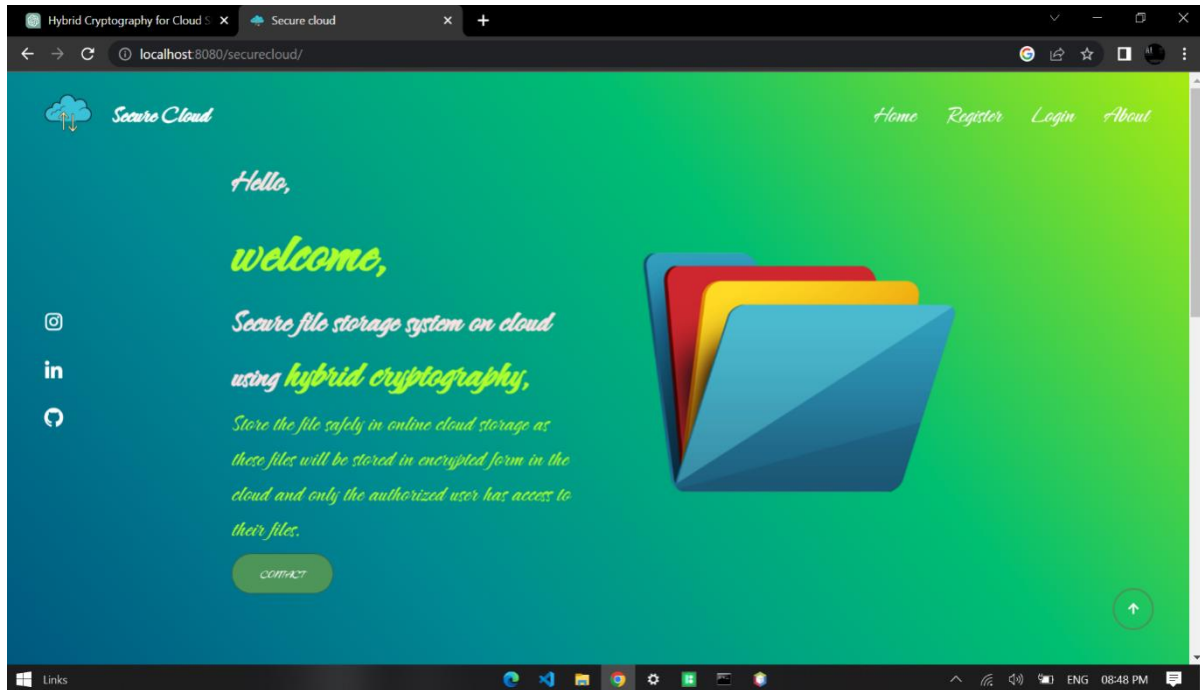


Fig 6.1 Home page

#### 6.2 Registration Page

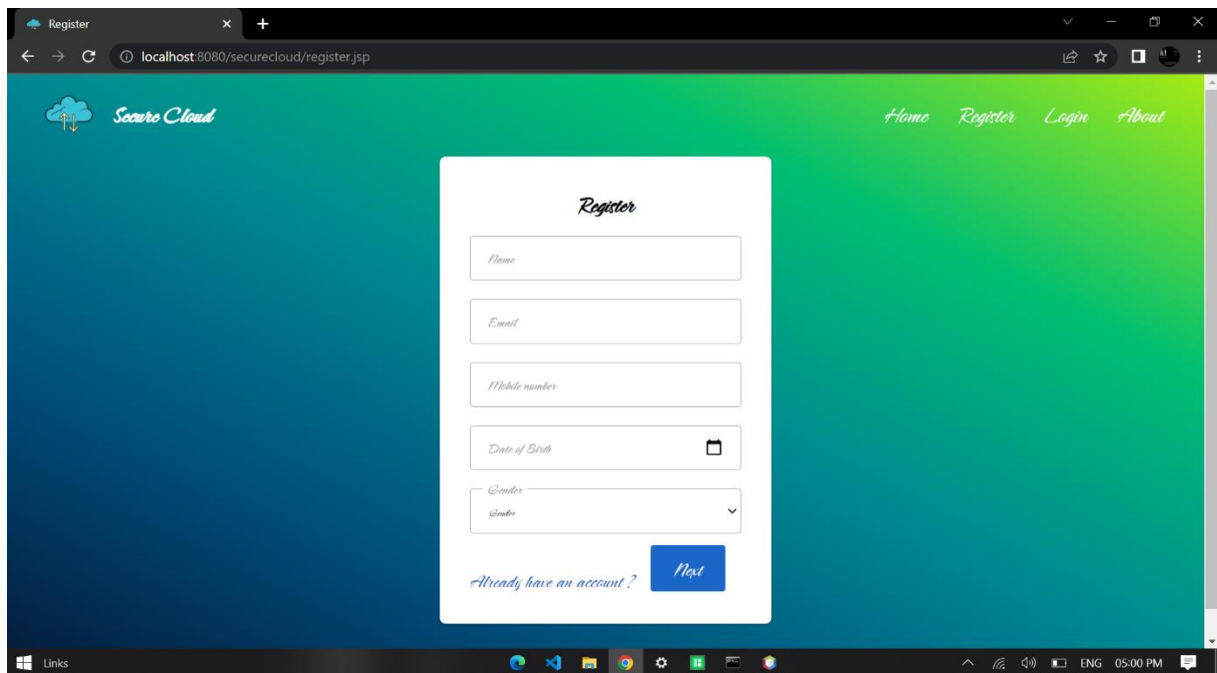


Fig 6.2 Registration Page

## 6.3 Key Exchange

The screenshot shows a web browser window with the address bar displaying `localhost:8080/securecloud/keyexchange.jsp?userid=100017`. The page has a green-to-blue gradient background. At the top left is the 'Secure Cloud' logo with a cloud icon. At the top right are links for 'Home', 'Register', 'Login', and 'About'. The main content is a white box titled 'Key Exchange'. Inside this box, there is a form with the following elements: a text input field labeled 'Your Registration id is:' containing the value '100017'; a text input field labeled 'your sample key is:' containing the value 'secret3212' with a blue 'copy' button to its right; a text input field labeled 'Enter your secret private key for ECDH key exchange:' containing the value 'secret3212'; and a blue 'Submit' button at the bottom.

Fig 6.3 Key Exchange

## 6.4 ECDH key agreement

The screenshot shows a web browser window with the address bar displaying `localhost:8080/securecloud/secretkey.jsp?userid=100017&secretkey=1E77D409FBC0A3E576A910C1FA03`. The page has the same green-to-blue gradient background and navigation links as Figure 6.3. The main content is a white box titled 'Generate User ID'. Inside this box, there is a form with the following elements: a red message line stating 'ECDH key agreement has been successfully generated.'; a text input field labeled 'Your ECDH key is:' containing the value '1E77D409FBC0A3E576A910C1FA03'; a text input field labeled 'OTP has been sent to your email id.'; and a blue 'Generate User ID' button at the bottom.

Fig 6.4 ECDH key agreement

## 6.5 User ID

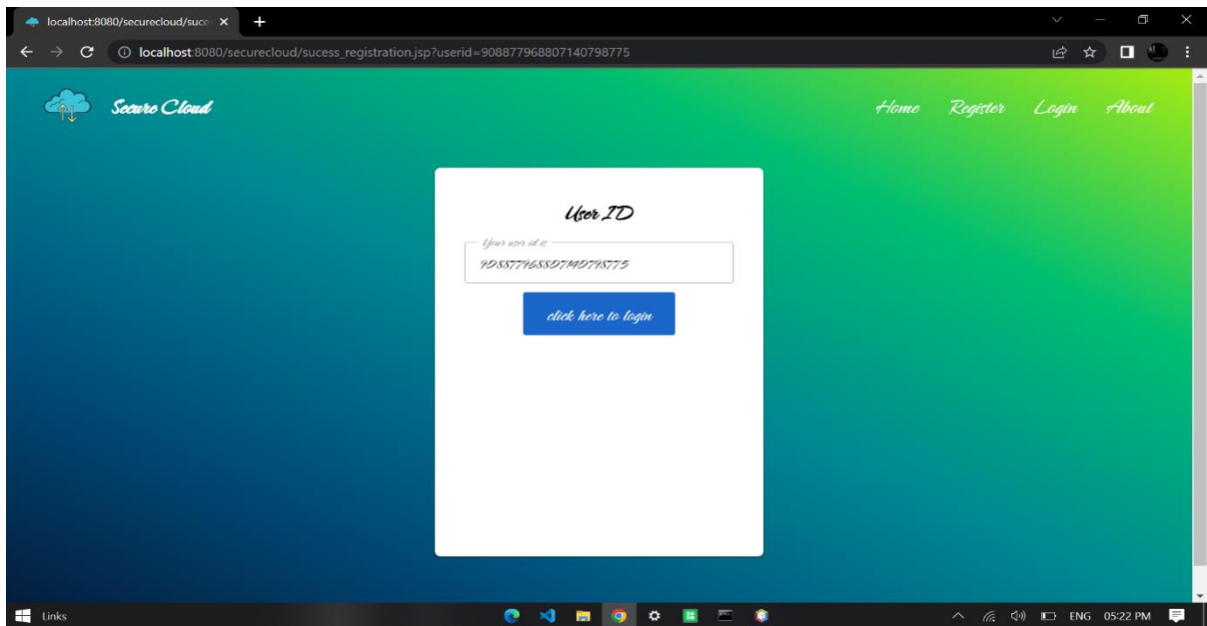


Fig 6.5 User ID

## 6.6 Login Page

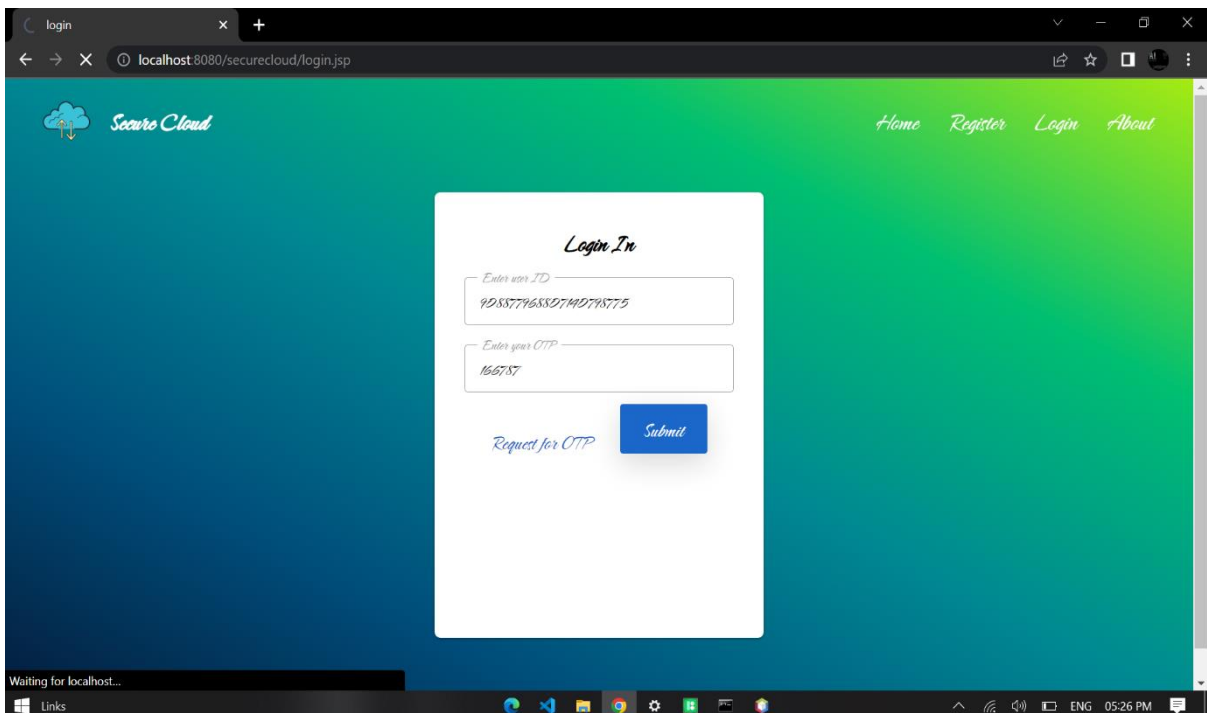


Fig 6.6 Login Page

## 6.7 User account page

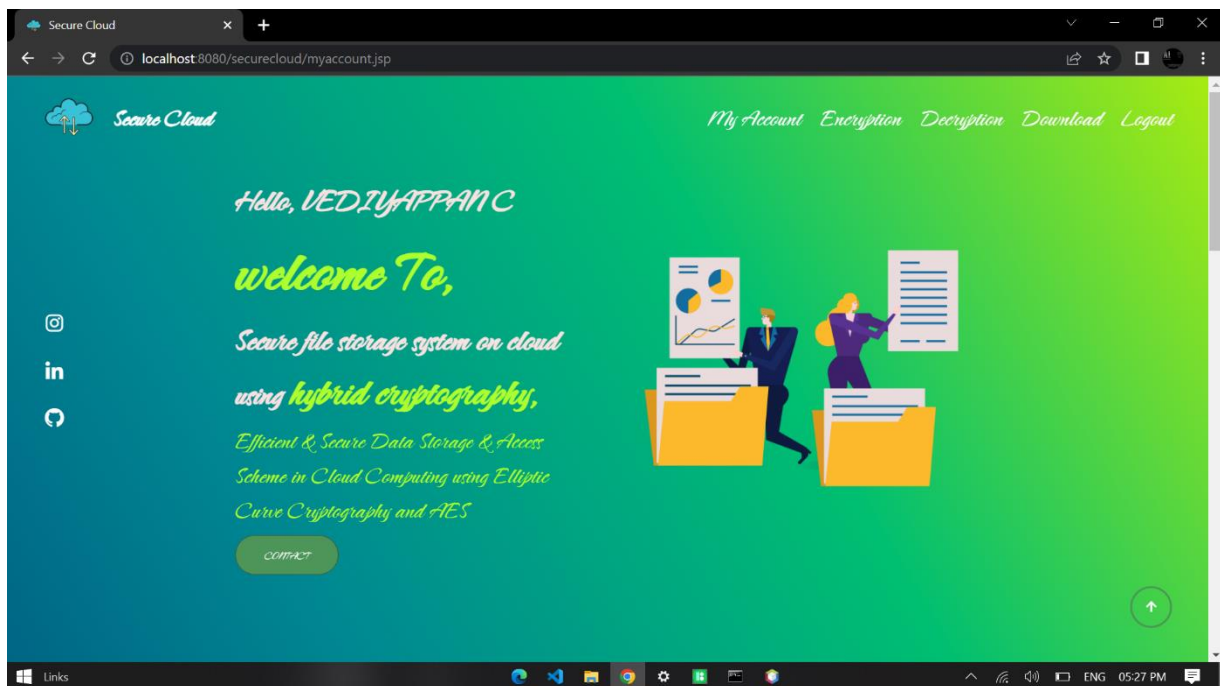


Fig 6.7 User account page

## 6.8 File Encryption

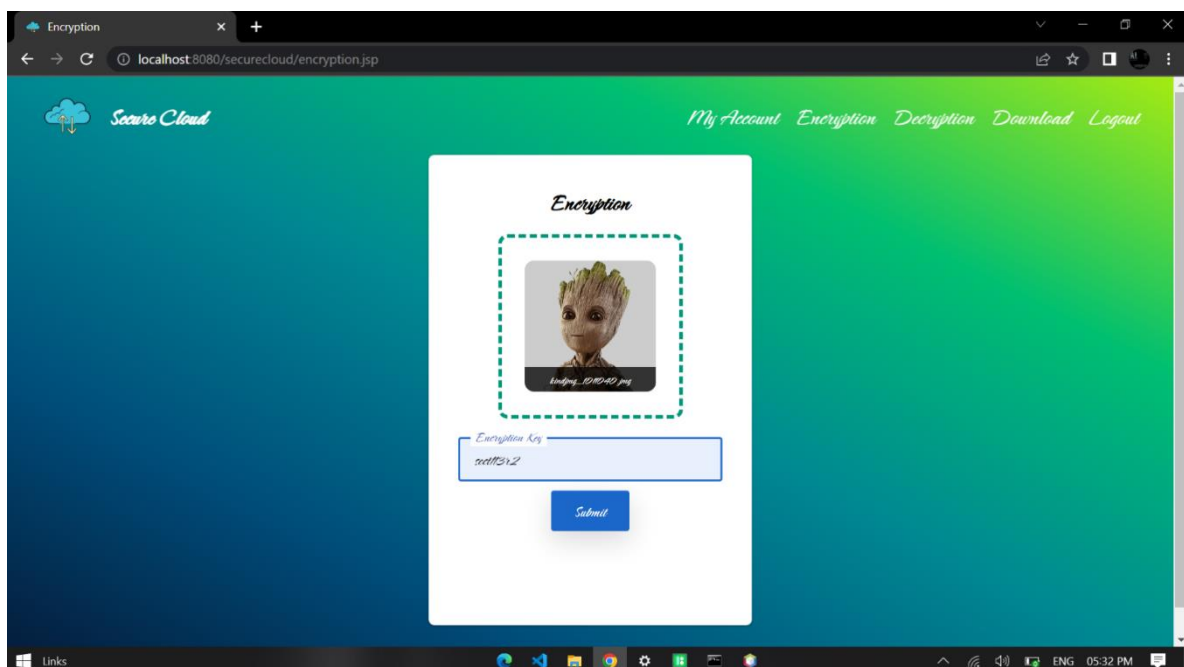


Fig 6.8 File Encryption

## 6.9 File Decryption

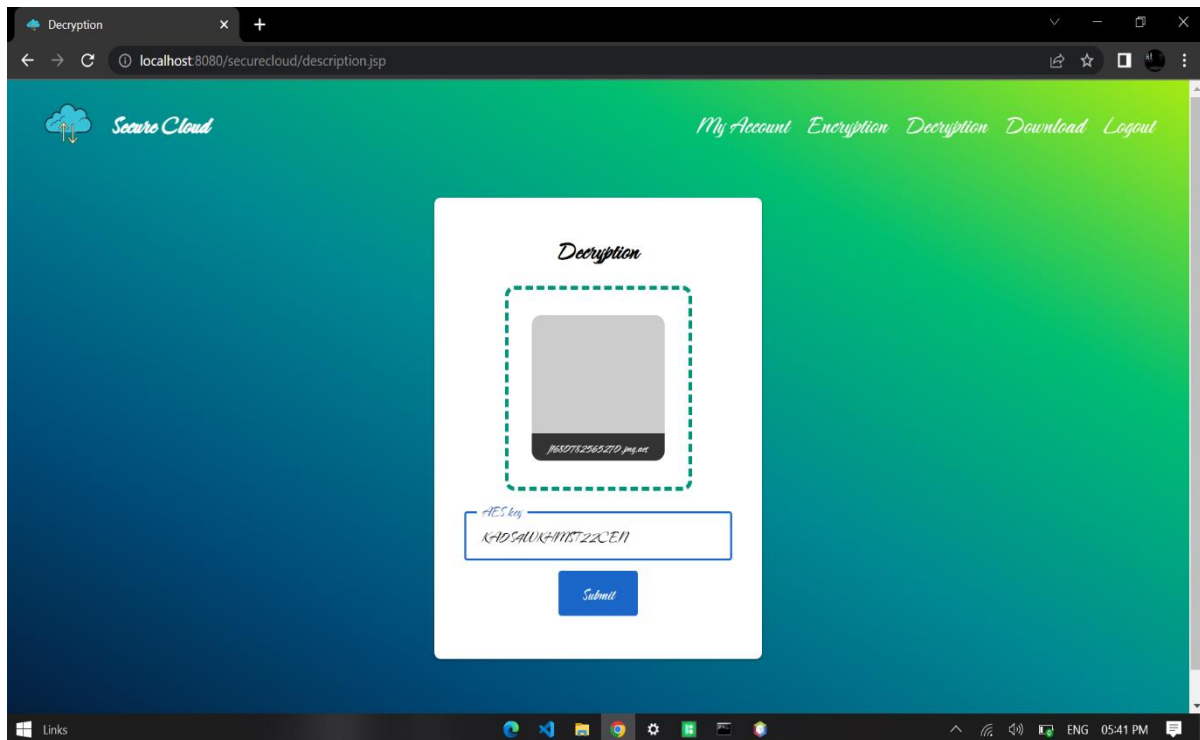


Fig 6.9 File Decryption

## 6.10 File Download after decryption

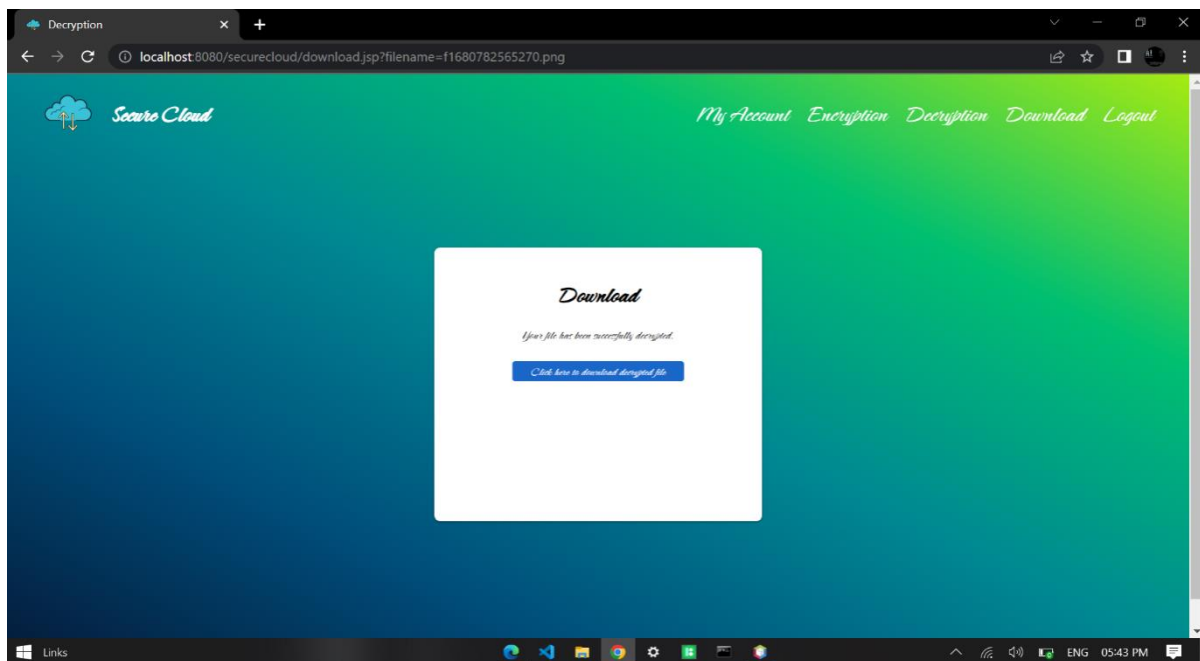


Fig 6.10 File Download after decryption

# **CHAPTER 7**

## **APPENDICES**



## CHAPTER 7

### APPENDICES

#### Login.jsp

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>login</title>
<link href="css/register.css" rel="stylesheet" type="text/css">
<link rel="stylesheet"
href="https://unpkg.com/boxicons@latest/css/boxicons.min.css">
<link rel="shortcut icon" href="images/logo.png" type="image/x-icon">
<link rel="stylesheet" href="css/index.css">
<link rel="stylesheet" href="css/encryption.css">
<script type="text/javascript">
function validation() {
var userid = document.getElementById('userid');
if (userid.value.trim() == "") {
alert('Please enter your userid');
userid.focus();
return false;
}
var opt = document.getElementById('opt');
if (opt.value.trim() == "") {
alert('Please enter your opt');
opt.focus();
return false;
}
}
```

```

function otpgeneration() {
var userid = document.getElementById('userid');
if (userid.value.trim() == "") {
alert('Please enter your userid');
userid.focus();
return false;
} else {
var xmlhttpRequest = init();
function init() {
if (window.XMLHttpRequest) {
return new XMLHttpRequest();
} else if (window.ActiveXObject) {
return new ActiveXObject("Microsoft.XMLHTTP");
}
}
xmlhttpRequest.open("POST", "generateOTP?userid=" + escape(userid.value),
true);
xmlhttpRequest.onreadystatechange = processRequest;
xmlhttpRequest.send(null);
function processRequest() {
if (xmlhttpRequest.readyState == 4) {
if (xmlhttpRequest.status == 200) {
processResponse();
}
}
}

function processResponse() {
var xmlMessage = xmlhttpRequest.responseXML;
var result =
xmlMessage.getElementsByTagName("sname")[0].firstChild.nodeValue;
alert(result);
}
}
}
</script>
</head>

```

```

<body>
<header>
<div class="logo">

<span>Secure Cloud</span>
</div>
<div class="navbar">
<ul class="navlist">
<li><a href="index.jsp">&nbsp;Home&nbsp;</a></li>
<li><a href="register.jsp">&nbsp;Register&nbsp;</a></li>
<li><a href="login.jsp">&nbsp;Login&nbsp;</a></li>
<li><a href="about.jsp">&nbsp;About&nbsp;</a></li>
</ul>
</div>
<div class="bx bx-menu" id="menu-icon">
</div>
</header>
<div class="box">
<form method="post" action="login" class="form">
<h1 class="form-title">Login In</h1>
<%
String msg = null;
msg = (String) session.getAttribute("MSG");
if (msg != null) {
%>
<div class="error"><%=msg%></div>
<%
session.removeAttribute("MSG");
} else {
session.setAttribute("MSG", "");
}
%>
<div class="form-group">
<input type="text" name="userid" id="userid" placeholder="User ID"
class="form-control" required />
String result = "";
@Override

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException {
    HttpSession session = request.getSession(true);
    userid = request.getParameter("userid");
    otp = request.getParameter("otp");
    /*Self generated algorithm */
    BigInteger ciphertxt = new BigInteger(userid);
    byte[] byteArray2 = ciphertxt.toByteArray();
    String s2 = new String(byteArray2);
    userid = s2.substring(3, s2.length());
    try {
        con = connection.dbConnection.makeConnection();
        String query = "SELECT name,utype FROM users WHERE userid = '" + userid
+ "' AND u_status = '1'";
        pst = con.prepareStatement(query);
        rst = pst.executeQuery();
        if (rst.next()) {
            name = rst.getString(1);
            utype = rst.getString(2);
            //check otp is exists or not
            try {
                Date dNow = new Date();
                SimpleDateFormat ft = new SimpleDateFormat("yyyy-MM-dd");
                String curDate = ft.format(dNow);
                String mysql_query = "SELECT * FROM user_login WHERE userid = '" +
userid + "' AND otp = '" + otp + "' AND created = '" + curDate + "'";
                pst = con.prepareStatement(mysql_query);
                rs = pst.executeQuery();
                if (rs.next()) {
                    session.setAttribute("ID", userid);
                    session.setAttribute("NAME", name);
                    session.setAttribute("UTYPE", utype);
                    response.sendRedirect("myaccount.jsp");
                } else {
                    session.setAttribute("MSG", "OTP is wrong. Please try again.");
                    response.sendRedirect("login.jsp");
                }
            } catch (Exception e) {
                //handle exception
            }
        }
    } catch (Exception e) {
        //handle exception
    }
}

```

```

}
} catch (Exception e) {
}
} else {
session.setAttribute("MSG", "Userid does not exists.");
response.sendRedirect("login.jsp");
}
} catch (Exception e) {
session.setAttribute("MSG", "Userid does not exists.");
response.sendRedirect("login.jsp");}}}}

```

### **dbconnection.java**

```

package connection;

import java.sql.PreparedStatement;

import java.sql.Connection;

import java.sql.DriverManager;

public class dbConnection {

static Connection con = null;

PreparedStatement pst = null;

public static Connection makeConnection() {

try {

Class.forName("com.mysql.jdbc.Driver").newInstance();

con

=DriverManager.getConnection("jdbc:mysql://securecloud.cjkb13c3i1m0.ap-
northeast-1.rds.amazonaws.com:3306/securecloud", "admin", "Vedi9095");

//con = DriverManager.getConnection("securecloud.cjkb13c3i1m0.ap-
northeast-1.rds.amazonaws.com", "admin", "Vedi9095");

} catch (Exception e) {

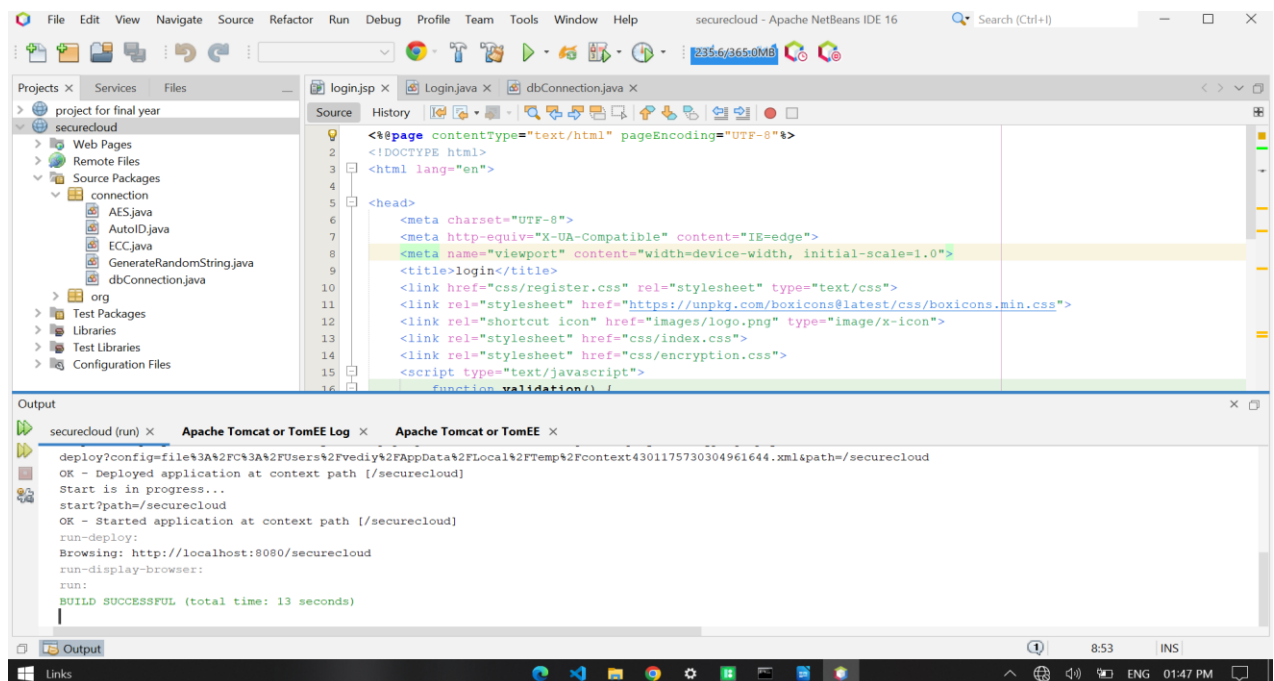
}

return con;

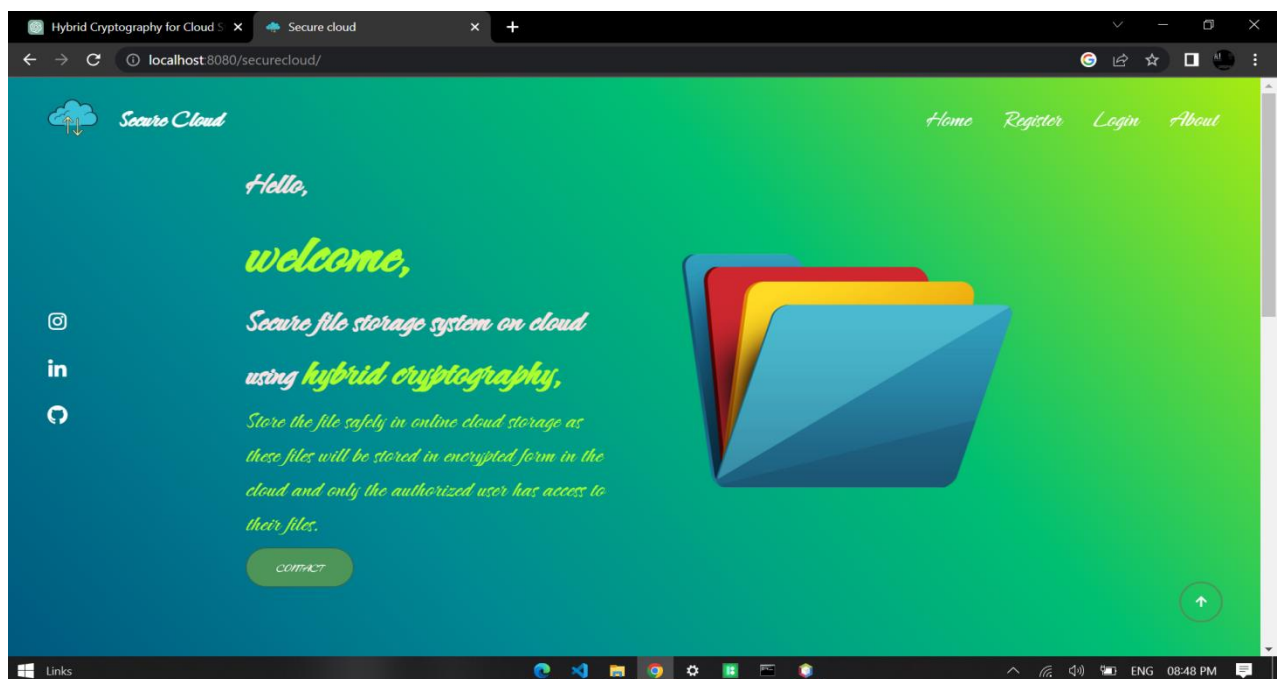
}

```

output:



## 7.1 Program execution



## 7.2 Home page

# **CHAPTER 8**

## **CONCLUSION AND FUTURE WORK**

## **CHAPTER 8**

### **CONCLUSION AND FUTURE WORK**

#### **8.1 CONCLUSION**

The data is stored at multiple places over the information space (over the Internet). It sounds similar to file hosting websites which stores the data that is being uploaded by different users and can be retrieved using proper authentication. The only difference is that the system for which project is targeted is an application based system like which will run on the client's own system. This application will allow users to upload file of different formats with security features including Encryption, secure OTP verification, uploading and downloading over the cloud securely.

In the last semester, we completed database modelling, new user registration module, secure key generation, exchange module along with introduction, user interface of the application and we managed to develop approximately 50% of the whole project. In current semester, we have worked on modules like, file encryption-decryption, secure file download and upload using user ID and another OTP, Secure AES Key generation for the file handling features, testing of the whole application, load balancing, checking the border cases to make sure that all the modules are completed and running.

This prototype works using a mixture of elliptical curve cryptography and symmetric key algorithm. ECC is used to achieve the process of user's verification and to keep the private data secure. AES algorithm is used to allow the user to store and access their data securely to the cloud by encrypting the data in the client side and decrypting the data after downloading from the cloud. Since the private key is owned only by the user of the data, no one can decrypt the data, even though the hacker can get the data through some approaches.



## 8.2 SCOPE FOR FUTURE WORK

The development of this project surely prompts many new areas of investigation. We have successfully completed all the phase of the development and did our research prior to the development about the various encryption techniques along with their strengths and weaknesses. After in depth comparisons and review we decided to use ECDH and AES Algorithms to implement the whole idea of this model. We referred multiple research papers whose links can be found in the references. Lastly, we implemented the model using Java Servlets & JSPs, Apache Tomcat server and MySQL for storage. We will be using cloud hosting services for the actual deployment of the project. For the GUI, we have used HTML and CSS 3.

In the last semester, we completed database modelling, new user registration module, secure key generation, exchange module along with introduction, user interface of the application and we managed to develop approximately 50% of the whole project. In current semester, we have worked on modules like, file encryption-decryption, secure file download and upload using user ID and another OTP, Secure AES Key generation for the file handling features, testing of the whole application, load balancing, checking the border cases to make sure that all the modules are completed and running.

Keeping all the deadlines in mind and the project development life-cycle of our project, we have been able to complete all the phases of the development cycle on time with minimum possible errors. Talking about scope of future work, we believe that it is possible to minimize the processing time taken in all the encryption-decryption processes by using professional hosting services and even better implementation of the security model. Also, some advanced hybrid cryptography algorithms and system can be incorporated to ensure cutting edge security and protection.

## REFERENCES

## REFERENCES

- 1) Arthur Rahumed, Henry C. H. Chen, Yang Tang, Patrick P. C. Lee, and John C. S. Lui “A Secure Cloud Backup System with Assured Deletion and Version Control” 2011 International Conference on Parallel Processing Workshops.
- 2) Ashutosh Kumar Dubey 1, Animesh Kumar Dubey 2, Mayank Namdev<sup>3</sup>, Shiv Shakti Shrivastava<sup>4</sup> “Cloud-User Security Based on R SA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment “in 2011.
- 3) Dr. K N Mishra, “A Novel Mechanism for Cloud Data Management in Distributed Environment. Data Intensive computing Applications for Big Data”, IOS Press,2018.
- 4) Eman M.Mohamed and Sherif EI-Etriby “Randomness Testing of Modem Encryption Techniques in Cloud Environment” in year 2008.
- 5) Qin Liu, Guojun Wang, and Jie Wu“Efficient Sharing of Secure Cloud Storage Services” 2010 .10th IEEE International Conference on Computer and Information Technology (CIT - 2010).
- 6) Uma Somani, Kanika Lakhani, Manish Mundra“Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing” 2010 IEEE 1st International Conference on Parallel, Distributed and Grid Computing (PDGC - 2010).
- 7) Xiang Tana, Bo Aib“The Issues of Cloud Computing Security in High-speed Railway “in 2011.