IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2

Secure File Storage System with Encryption and Role-Based Access

D. Sandhya Rani[1], Bejjanki Pooja[2], B. Aditya[3]
Banda Architha[4], Munige Sai Nikhil[5], Nagunuri Chandu[6]
Assistant Professor, Department of CSE[1,2,3]
UG Student, Department of CSE[4,5,6]
CMR Technical Campus, Hyderabad, Telanagana, India
davu.sandhya@gmail.com, poojareddybejjanki@gmail.com, Adi.sacs@gmail.com
archithabanda2005@gmail.com, sainikhilbts7@gmail.com, chandunagunuri6353@gmail.com

Abstract: *We live in a world where people share a lot of information online. This is why it is very important to keep our files safe and only let the right people see them. The Secure File Storage System project is about building a system that protects our data with encryption and makes sure only the right people can access it. Users can upload, store and get to their files safely. The Secure File Storage System gives people roles like Admin, Manager or Employee. They can only do things that their role allows. We use an encryption method called AES-256 to keep the files safe. We also use HTTPS to protect the data when it is being sent. The Secure File Storage System only lets people do what they need to do which helps keep everything safe. It also has a panel for the admin to manage users, roles and see what people are doing. All the things people do on the Secure File Storage System like logging in and getting to files are recorded so we can see what is happening and keep everything secure.*

Keywords*:* *files safely*.

I. INTRODUCTION

These days we share a lot of data online. People and companies use cloud and web platforms to store files like money records, personal documents and business information. This also has risks like data breaches people getting to things they should not and cyberattacks. If we do not protect our information it can be used by people who should not have it which can cause financial losses and hurt our reputation. That is why we need a file storage system that keeps our data safe and only lets the right people get to it.

One of the ways to keep our data safe is by using encryption. It makes sure that even if someone gets to our files without permission they will not be able to read or use the information. The Advanced Encryption Standard, AES-256 is one of the most reliable encryption methods used today. By using AES encryption in file storage systems our data stays protected when it is stored and when it is being sent. Encrypting files before saving them on the server adds a layer of security which helps prevent data leaks even if the system is hacked or storage devices are stolen.

Along with encryption controlling who can access our data is also very important for keeping it secure. This is done by giving users access based on their roles and responsibilities. Of giving everyone the same access administrators can create roles like Admin, Manager or Employee and give them specific permissions. This way users can only use the files. Features they actually need. For example an admin might have control over the system while an employee may only be able to see documents. This helps prevent misuse of data reduces the chances of threats and avoids leaks.

The Secure File Storage System project is about building a file storage system by combining encryption with role-based access control. Users will log in safely to the Secure File Storage System. Their activities will be tracked to make sure everything is accountable. Every file uploaded will be encrypted using AES-256 before being stored and access to these files will depend on the users assigned role.

By protecting against both attacks and inside misuse the Secure File Storage System offers a solution for safely managing sensitive data. It can be useful for businesses, schools and any organization that needs to handle information

II. PROBLEM STATEMENT

A. Weak Data Security and Limited User Control

Many existing file storage systems do not protect our data well. In cases encryption is handled by the service provider, which means users have little control over how their data is secured. This becomes risky especially if the system is compromised as sensitive information can be exposed or misused.

Systems usually follow permission models that do not consider user roles. Because of this users may access files that're not relevant to their responsibilities. This increases the chances of both misuse and accidental data leaks within an organization.

Lack of Proper Monitoring and Integrated Security

Another issue is the lack of tracking and monitoring. Many systems do not maintain activity logs making it difficult to identify actions or investigate security issues. In addition security features like encryption and access control are often not well connected, which creates gaps in protection.

III . RELATED WORK

Several researchers have worked on improving data security in file storage systems using encryption and access control techniques. Many studies focus on the use of the Advanced Encryption Standard (AES) to protect data. AES is widely accepted as an encryption method that ensures data confidentiality and protects against unauthorized access. These works highlight that encrypting data before storage is a way to prevent data breaches.

Another important area of research is Role-Based Access Control (RBAC) which helps manage user permissions based on their roles within an organization. Early research in this field introduced models where access rights are assigned according to responsibilities making systems more organized and secure. RBAC has been widely adopted because it simplifies access management and reduces the risk of access.

Some studies combine both encryption and access control to improve system security. For example research on storage systems shows that integrating AES encryption with RBAC can significantly enhance data protection while maintaining efficient access control. In addition audit mechanisms and logging have been suggested to improve transparency and accountability in systems.

There is also work focusing on security models that use techniques, such as combining AES with other algorithms or implementing advanced key management strategies. These approaches aim to strengthen security by addressing limitations in methods.

Overall existing research clearly shows that both encryption and access control are essential for data storage. However many systems still treat these features separately. This creates a need for solutions that combine encryption with structured access control and proper monitoring, which is the focus of the Secure File Storage System project.

IV. METHODOLOGY

A. User Authentication

The Secure File Storage System starts with a login process where users register and access the platform using their credentials. Passwords are safely stored using encryption or hashing techniques to prevent access. Once logged in each user is assigned a role such as Admin, Manager or Employee which defines their level of access in the Secure File Storage System.

B. Role-Based Access Control (RBAC)

After authentication the Secure File Storage System applies role-based access control to manage permissions. Of giving equal access to all users permissions are assigned based on their roles. For example an Admin can manage users and system settings while an Employee can only access their files. This ensures that users can only interact with data that's relevant to them.

C. File Encryption Process

Whenever a user uploads a file it is automatically encrypted using AES-256 before being stored. This ensures that the file remains secure and unreadable to users. Even if someone gains access to the storage system the encrypted data cannot be understood without the key.

D. Secure File Storage

Once encrypted files are stored on a server along with details such as file name, owner and upload time. Storing files in encrypted form adds a layer of protection. Ensures data safety even in case of system compromise.

E. File Access and Decryption

When a user requests to access a file the Secure File Storage System first checks their role and permissions. If the user is authorized the file is decrypted securely. Then made available, for download or viewing. This step ensures that only valid users can access data.

The Secure File Storage System also uses Role-Based Access Control to manage user permissions. This means that not all When a user asks for a file the system checks if they are allowed to see it. The system does this by looking at the rules that say who can see what. If the user is allowed to see the file the system gets the file from where it's stored. The file is then sent to a part of the system that can decrypt it. The file is changed back to how it was before it is given to the user. The system also keeps track of what users do. This helps the people in charge of the system make sure everything is working correctly and that users are not doing anything they should not be doing.

The way the system is built is like a team working together. Each part of the system does its job to make sure the files are stored safely and can be gotten when they are needed.'

V. PROPOSED SYSTEM

The proposed system helps to store and manage files safely and efficiently. It does this by using encryption and controlling who can access the files. The goal is to fix the problems with systems by keeping data protected at all times while still being easy for authorized users to access.The system has a way to check who users are.

Users must sign up. Log in with their correct details. This makes sure verified people can use the system. Each user has a job, like Admin, Manager or Employee.

This job decides what they can. Cannot do in the system.It helps to keep things in order and stops people from doing things they should not.One of the features of the system is that it uses AES-256 encryption for all files. When a file is uploaded it gets encrypted away before being stored on the server.

This means that even if someone gets access to the storage system they cannot read the data without the right decryption key.This approach keeps information safe and reduces the risk of data breaches.The system also uses Role-Based Access Control (RBAC) to manage what users can do.Of giving all users the same access it gives permissions based on their jobs.

For example an Admin can control users and system settings.

A Manager can. Manage files for their department.

An Employee can only access files assigned to them.

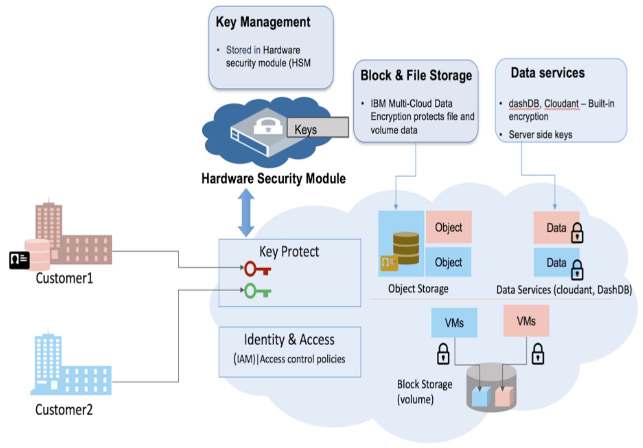
This makes sure users only interact with data that's relevant to their work. It reduces the chances of misuse. The system keeps a record of all user activities. This includes login attempts, file uploads, downloads and sharing. These records have timestamps. They help administrators monitor the system detect behavior and investigate security incidents.

The system also keeps data safe when it is being transferred. It uses HTTPS protocols for all communication between users and the system. This prevents risks like data interception or man-in-the-middle attacks.

The system is designed to handle users and a lot of data. It does this without slowing down. The interface is simple and easy to use. Users can upload, access and manage files without needing expertise.
Overall the proposed system provides a solution for secure file storage. It combines encryption, access control and monitoring into one platform. It ensures data confidentiality, controlled accessibility and accountability. This makes it suitable, for organizations that handle information.
The system helps to store and manage files safely and efficiently. It uses encryption and controls who can access the files. The system keeps data protected at all times while still being easy for authorized users to access. It is designed to handle users and a lot of data without slowing down.

VI. SYSTEM ARCHITECTURE

The file storage system architecture is designed to be simple and structured. This ensures both security and smooth functioning.


Key Management
• Stored in Hardware security module (HSM
Block & File Storage
• IBM Multi-Cloud Data Encryption protects file and volume data
Data services
• dashDB, Cloudant – Built-in encryption
• Server side keys
Keys
Hardware Security Module
Customer1
Key Protect
Object
Object
Object Storage
Data
Data
Data Services (cloudant, DashDB)
Identity & Access
(IAM)|Access control policies
VMs
VMs
Block Storage
(volume)
Customer2

The system mainly uses a client-server model. Users interact with the system through a web interface. All processing happens on the server side.

- ➢ At the user end the client (browser or application) is used to log in.
- ➢ Users can upload files. Request access to stored data.
- ➢ All user requests are sent securely to the server through HTTPS.
- ➢ This protects the data during transmission.

The server side is divided into modules. The authentication module checks who the user is.It gives users roles like Admin, Manager or Employee. Once the user is authenticated the access control module (RBAC) checks if they can do an action.
When a file is uploaded it goes to the encryption module. Here it is encrypted using AES-256.The encrypted file is then stored. Details like file name, owner and access permissions are stored in a database.
When a user requests a file the system checks permissions. If access is allowed the file is. Sent to the decryption module. Here it is converted back to its form. Then it is delivered to the user. The system also has a logging module.This records all user activities, like logins, uploads and downloads.
This helps in monitoring the system. It also helps in keeping track of what users do. Overall the architecture uses an approach. Each module works together. The modules are authentication, access control, encryption, storage and logging. They provide an efficient file storage system.

VII. SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS:
System: Pentium IV 2.4 GHz.
Hard Disk: 40 GB.
Ram: 512 Mb.

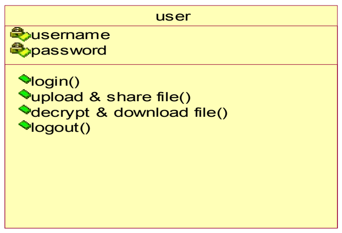
B. SOFTWARE REQUIREMENTS:
Operating system: Windows.
Coding Language: python.

VIII. UML DIAGRAMS

We use diagrams to show how the system works. These diagrams help us understand how the different parts of the system work together. In this project we used these diagrams to explain how the system is designed and how it works.

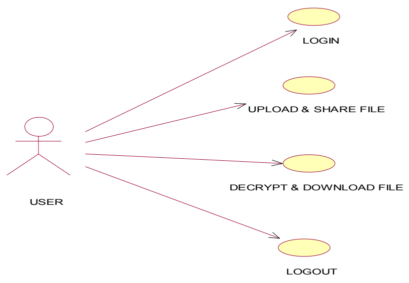
A. Use Case Diagram

The use case diagram shows how users interact with the system. It shows what different users can do like log in upload files download files and manage users.



B. Class Diagram

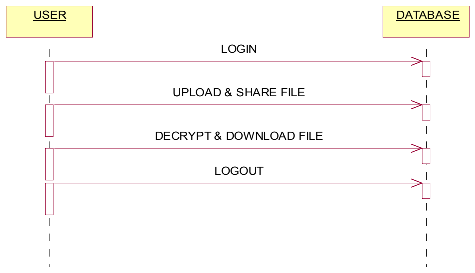
The class diagram shows the structure of the system. It shows the parts of the system what they do and how they are connected



C. Sequence Diagram

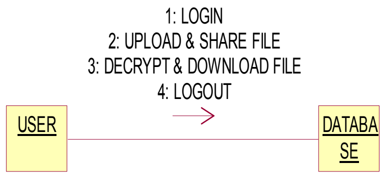
The sequence diagram shows how the different parts of the system work together. It shows what happens when a user uploads or downloads a file.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2



D. Collaboration Diagram

The collaboration diagram shows how the different parts of the system work together to get something done. It shows how they talk to each share jobs.



IX. IMPLEMENTATION

We built the system step by step. This way we made sure it is safe and easy to use. Each part of the system has a job and all the parts work together to keep the files safe and make it easy to manage them.

A. User Authentication

The system starts with users logging in and creating accounts. Users log in with their names and passwords. To keep the accounts safe passwords are stored in a way. When a user logs in they are given a role, like administrator, manager or employee. This role decides what they can do in the system.

B. Role-Based Access Control

After a user logs in the system decides what they can see based on their role. Not everyone can see everything. For example an administrator can manage users and settings. An employee can only see their own files or files that are shared with them.

C. File Decryption

When a user uploads a file the system encrypts it automatically. This means the file is changed into a code that only the right users can understand. When a user wants to download the file it is changed back to how it was

D. File Storage

The encrypted files are stored safely on the server. The system also stores information about the files like their names, who uploaded them and when they were uploaded. This helps the system find the files when they are needed.

E. Activity Logging

The system keeps track of what users do like logging in uploading files and downloading files. This helps the people in charge of the system make sure everything is working correctly and that users are not doing anything they should not be doing.

F. User Interface

The system is designed to be simple and easy to use. Users can upload, download and manage files without getting confused. Administrators also have a page to control users and permissions easily.

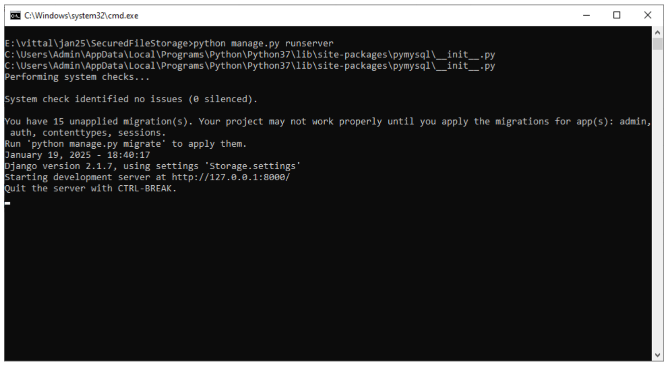
G. Overall Working

All these parts of the system work together to create a place to store files. The system not keeps the files safe but also makes sure that only the right users can see them and it does all this in a simple and efficient way.

XI. RESULT

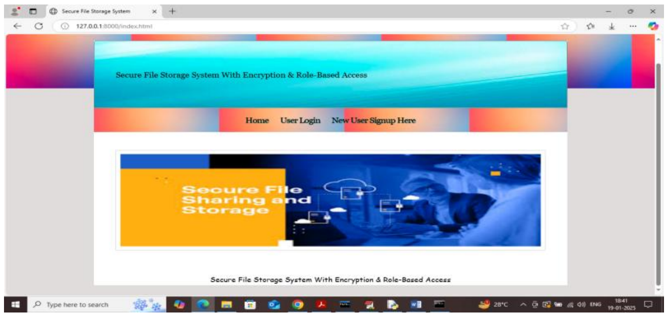
We tested the system to see how well it works in terms of safety, performance and ease of use. The results were good. The system worked as expected. All the main features worked correctly. Users could log in upload and download files and see the files they were allowed to see based on their roles.
The encryption part of the system worked well. Every file was encrypted before it was stored and only decrypted when the right user accessed it. This means the files are safe even if someone tries to access them without permission.
The system was also easy to use. Users could navigate through the pages easily. Uploading and downloading files was smooth and quick. They did not have any problems using the system.
The system also kept track of what users did like logging in and accessing files.
This helps the people in charge of the system make sure everything is working correctly and that users are not doing anything they should not be doing.
Overall the system was safe, reliable and user-friendly making it good for storing files.

X. IMPLEMENTED OUTPUT

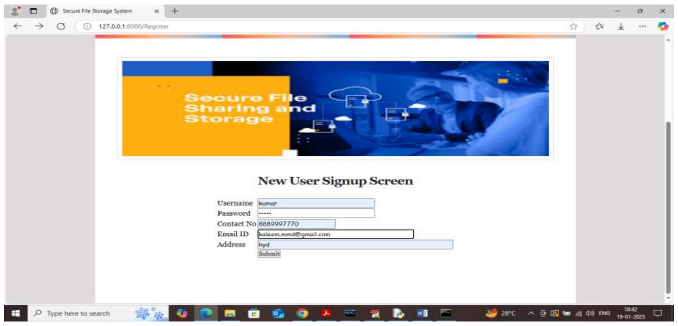


```
C:\Windows\system32\cmd.exe

E:\vittal\jan25\SecuredFileStorage>python manage.py runserver
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 19, 2025 - 18:40:17
Django version 2.1.7, using settings 'Storage.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

The python server is started now. Open a browser. Type the URL as http://127.0.0.1:8000/index.html. Then press the enter key to see the page.


Secure File Storage System
127.0.0.1:8000/index.html
Secure File Storage System With Encryption & Role-Based Access
Home
User Login
New User Signup Here
Secure File Sharing and Storage
Secure File Storage System With Encryption & Role-Based Access
Type here to search
28°C
ENG
18:41
19-01-2025

On this page click on the 'New User Sign up' link to get to the page.

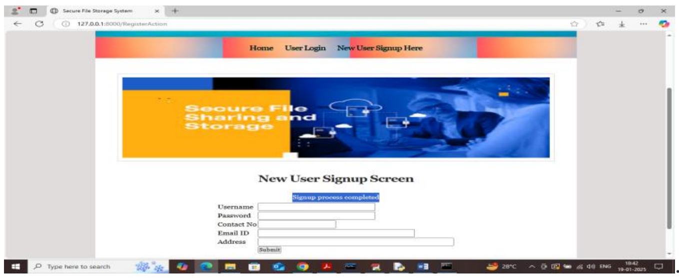


New User Signup Screen

Username	kumar
Password	•••••
Contact No	8889997770
Email ID	kaleem.mmd@gmail.com
Address	hyd

Submit

The user is entering sign up details Above. Giving a valid email address to receive OTP emails. Then they press the button to get to the page



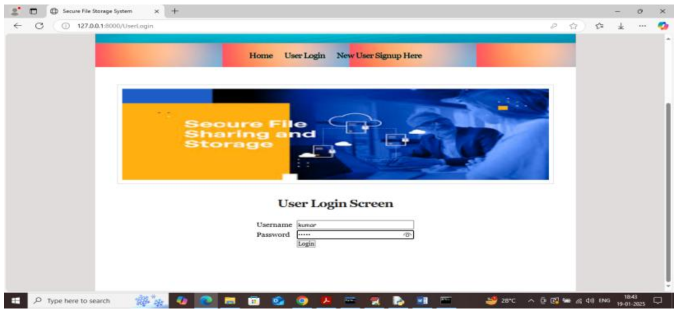
New User Signup Screen

Signup process completed

Username	
Password	
Contact No	
Email ID	
Address	

Submit

The user sign up is completed now. You can add users like this. Click on the 'User Login' link to get to the page.

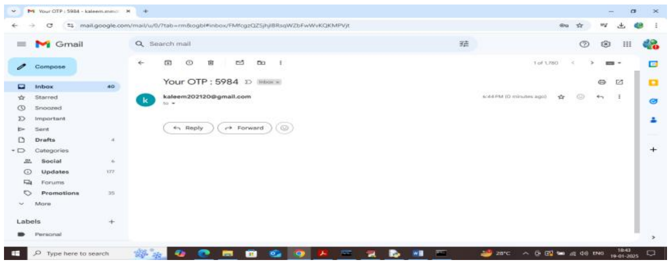


User Login Screen

Username	kumar
Password	•••••

Login

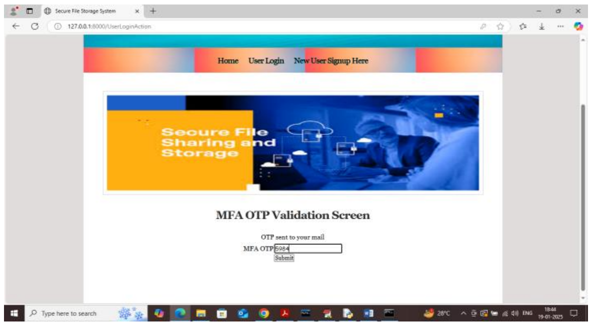
The user is logging in now. After logging in they will get an OTP in their email like this.



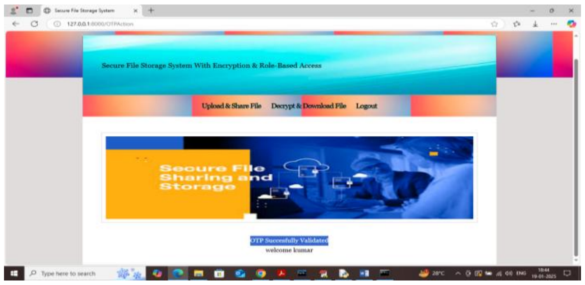
Your OTP : 5984

kaleem202120@gmail.com

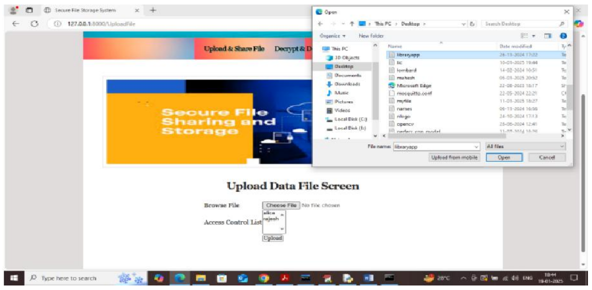
The user received the OTP in their email. Now they need to validate it in the application like this.


Home User Login New User Signup Here
Secure File Sharing and Storage
MFA OTP Validation Screen
OTP sent to your mail
MFA OTP
Submit

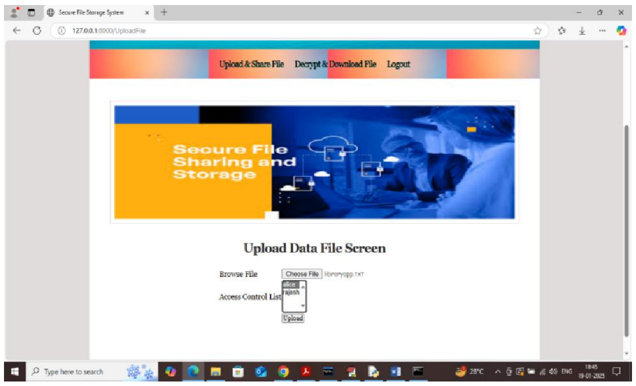
The user is validating the OTP now. Then they press the button to get to the page.


Secure File Storage System With Encryption & Role-Based Access
Upload & Share File Decrypt & Download File Logout
Secure File Sharing and Storage
OTP Successfully Validated
welcome kumar

Click on the 'Upload & Share File' link to upload a file and get to the page.

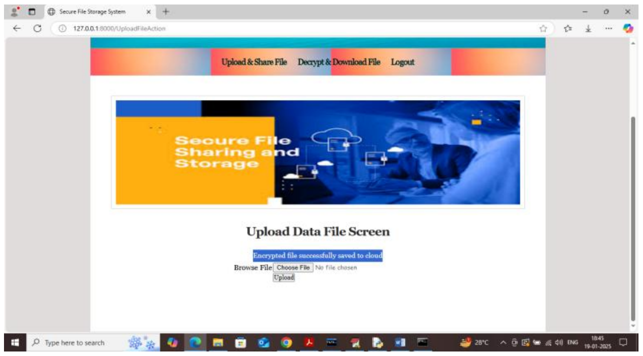


Select a file. Upload it. Choose the users you want to share it with by holding the CTRL key if you want to share with users. Then press the button to save the file and get to the page.


Upload & Share File Decrypt & Download File Logout
Secure File Sharing and Storage
Upload Data File Screen
Browse File
Choose File
Access Control List
alice
rajesh
Upload

Select the users. Then press the button to get to the next page. Give share permission role to Alice.

ISSN: 2581-9429 Volume 6, Issue 4, March 2026 Impact Factor: 8.2

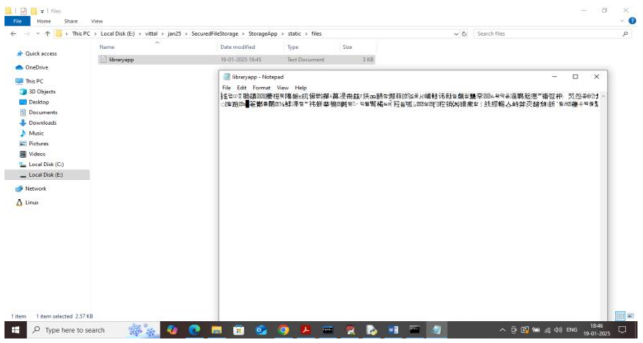


Upload Data File Screen

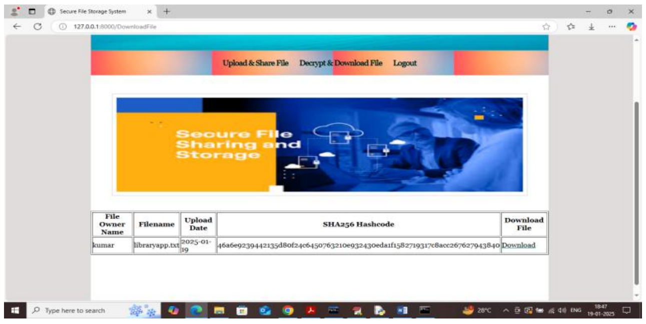
Encrypted file successfully saved to cloud

Browse File Choose File No file chosen
Upload

The encrypted file is saved in the cloud now. You can see it in the screen.

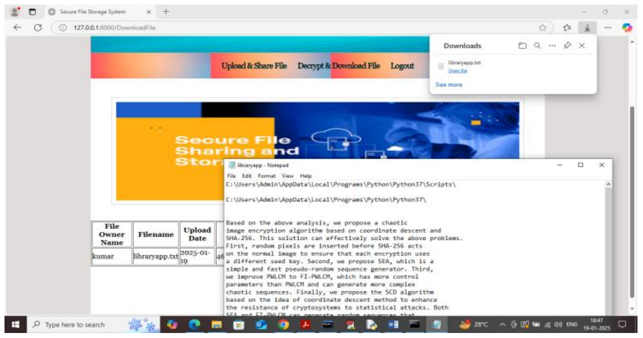


The uploaded file can be seen in encrypted format. All encrypted files will get saved inside the 'StorageApp//files folder. You can upload as files as you want now. Click on the 'Decrypt & Download File' to get a list of files.



File Owner Name	Filename	Upload Date	SHA256 Hashcode	Download File
kumar	libraryapp.txt	2025-01-19	46a6e9239442135d80f24c6450763210e932430eda1f1582719317c8acc267627943840	Download

You can see a list of shared files along with the SHA256 hash code now. Click on the 'Download' link to download the file and get to the page.

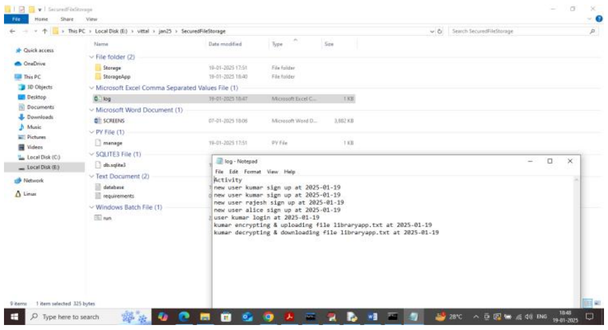


```
C:\Users\Admin\AppData\Local\Programs\Python\Python37\Scripts\

C:\Users\Admin\AppData\Local\Programs\Python\Python37\


Based on the above analysis, we propose a chaotic
image encryption algorithm based on coordinate descent and
SHA-256. This solution can effectively solve the above problems.
First, random pixels are inserted before SHA-256 acts
on the normal image to ensure that each encryption uses
a different seed key. Second, we propose SEA, which is a
simple and fast pseudo-random sequence generator. Third,
we improve PWLCM to FI-PWLCM, which has more control
parameters than PWLCM and can generate more complex
chaotic sequences. Finally, we propose the SCD algorithm
based on the idea of coordinate descent method to enhance
the resistance of cryptosystems to statistical attacks. Both
```

The file is downloaded in decrypted format now.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, March 2026
Impact Factor: 8.2

You can. Download as many files as you want, by following these steps.



```
Activity
new user kumar sign up at 2025-01-19
new user kumar sign up at 2025-01-19
new user rajesh sign up at 2025-01-19
new user alice sign up at 2025-01-19
user kumar login at 2025-01-19
kumar encrypting & uploading file libraryapp.txt at 2025-01-19
kumar decrypting & downloading file libraryapp.txt at 2025-01-19
```

You can see all log details saved inside the log.csv file now. This file contains all the details of user activities.

XII. CONCLUSION

In this project we built an reliable system to store files. By using encryption and role-based access control the system makes sure that sensitive files are safe and can only be seen by the users. The system is also simple and easy to use making it good for real-world applications.

The use of encryption helps protect files from unauthorized access and role-based access control makes sure that users can only see or manage files that are relevant to their jobs. The system also has features like login, activity logging and safe communication. These features make the system more secure.

The system was designed to be simple and user-friendly allowing users to easily upload, access and manage files. The system is easy to use and maintain a level of safety and control. This makes it suitable for real-world applications.

Overall this project provides a solution for file storage by balancing safety, usability and efficiency. The project provides a solution for file storage. The file storage system is very efficient.

XIII. FUTURE WORK

While the current system provides safety and performance there is always room for improvement. The system can be made secure by adding extra layers of security during login. The system can also be improved by adding features like artificial intelligence-based threat detection. This will help identify activities in real-time.

The system can also be extended to support users and devices making it more efficient. An application can be developed to make the system more accessible and user-friendly. Better ways of managing keys and backing up files can be implemented to strengthen data protection.

Overall these improvements can make the system more secure, flexible and suitable for large-scale applications.

REFERENCES

[1]. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. This book provides information about encryption algorithms, including the one used in the project.

[2]. Sandhu, R., Coyne, E. J., Feinstein, H. L., & Youman C. E. (1996). Role-Based Access Control Models. This paper is about role-based access control concepts and implementation.

[3]. Menezes A. J., van Oorschot, P. C. & Vanstone S. A. (1996). Handbook of Applied Cryptography. This book is a reference for encryption techniques and security protocols.

[4]. Bishop, M. (2003). Computer Security: Art and Science. This book discusses secure system design principles and audit logging mechanisms.

[5]. National Institute of Standards and Technology (NIST). (2001). Specification for the Advanced Encryption Standard (AES). This publication details the AES encryption standard.

[6]. Sandhu, R. S., & Ferraiolo, D. F. (1999). Access Control and Attribute-Based Access Control. This paper examines advanced access control models.

[7]. OWASP Foundation. (2023). OWASP Top Ten Security Risks. This publication provides web application security guidelines.

[8]. Kshetri, N. (2017). Blockchain's roles in strengthening cybersecurity and protecting privacy. This reference is about the use of blockchain in audit and data integrity.

[9]. Dworkin, M. (2001). Recommendation for Block Cipher Modes of Operation. This publication provides guidelines, for encryption modes to file encryption