SECURE FILE STORAGE SYSTEM WITH ENCRYPTION AND ROLE-BASED ACCESS

NAVIRI RAJU

Reg. No. 24Q71F0038

naviriraju123@gmail.com

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY(Autonomous)

Under the guidance of Mrs A .ANITHA MTech

anithaadireddy96@gmail.com

***Abstract*—*The increasing reliance on cloud storage services has raised significant concerns regarding data confidentiality, integrity, and unauthorized access. This project proposes a secure cloud-based file storage system that addresses these concerns by implementing client-side encryption, strong cryptographic algorithms, and role-based access control. Files are encrypted locally using AES and RSA before being uploaded to the cloud, ensuring that plaintext data never leaves the user's environment. The system architecture follows a zero-knowledge model, integrating secure authentication, key management, and encrypted communication protocols. Technologies such as Python, Flask, FastAPI, React, and cloud storage platforms like AWS S3 are used to implement the system. Testing confirms reliable performance, accurate encryption and decryption, secure cloud operations, and a user-friendly interface. The system enhances trust and privacy in cloud environments while maintaining usability and scalability.*

Keywords*—*Cloud Storage Security; Client-Side Encryption; Role-Based Access Control; AES Encryption; RSA Encryption.

I. INTRODUCTION

Cloud storage services have become integral to modern data management, offering scalability, accessibility, and cost-efficiency. However, the reliance on third-party providers raises concerns about data confidentiality, integrity, and unauthorized access. Traditional cloud storage systems often rely on server-side encryption managed by the provider, which may expose data to insider threats and breaches. This project introduces a secure file storage system that ensures end-to-end encryption, user-controlled key management, and role-based access to enhance data privacy and security.

The proposed system encrypts files locally on the user's device using strong cryptographic algorithms such as AES and RSA. This approach ensures that plaintext data never leaves the user's environment and is only accessible to authenticated users with valid decryption keys. The system integrates secure communication protocols, user authentication, and key management mechanisms to provide a robust and user-friendly solution for secure cloud storage.

II. LITERATURE SURVEY

Several studies have explored security challenges in cloud computing and proposed various solutions to address them. Subashini and Kavitha [1] conducted a survey on security issues in cloud service delivery models, highlighting the importance of encryption and access control. Ren et al. [2] discussed security challenges specific to public cloud environments, emphasizing the need for client-side encryption and user-controlled key management.

Armbrust et al. [3] provided an overview of cloud computing, outlining its benefits and potential risks. Stallings [4] detailed cryptographic techniques and network security principles that form the foundation of secure cloud storage systems. Li et al. [5] proposed attribute-based encryption for secure sharing of personal health records in cloud environments, demonstrating the effectiveness of advanced cryptographic methods.

Sahai and Waters [6] introduced fuzzy identity-based encryption, enabling flexible access control based on user attributes. Boneh and Franklin [7] presented identity-based encryption using the Weil pairing, offering a novel approach to key management. Yang et al. [8] developed DAC-MACS, a data access control scheme for multi-authority cloud storage systems, addressing the challenges of decentralized access control.

Mell and Grance [9] defined cloud computing and outlined its essential characteristics, service models, and deployment models. Buyya et al. [10] discussed the vision and reality of cloud computing as the fifth utility, emphasizing its potential impact on IT platforms. These studies provide valuable insights into the design and implementation of secure cloud storage systems, informing the development of the proposed solution.

TABLE I. LITERATURE SURVEY TABLE

Table No.	CONTENT	Page No.
2. I	Literature survey table	8

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

In existing cloud storage systems, users typically upload files directly to cloud servers, relying on the security mechanisms provided by the cloud service provider. While these platforms implement basic protections such as access control, user authentication, and server-side encryption, they do not guarantee complete confidentiality. In many cases, the cloud provider holds the encryption keys, meaning that data can potentially be accessed by insiders, compromised accounts, or malicious attackers if the server is breached.

Additionally, most existing systems lack client-side encryption, where files are secured before leaving the user's device. This exposes data during transmission and makes it vulnerable to man-in-the-middle attacks, unauthorized modifications, and privacy violations. Due to limited user control over encryption and key management, traditional cloud storage solutions are insufficient for storing highly sensitive or confidential information.

B. Proposed System

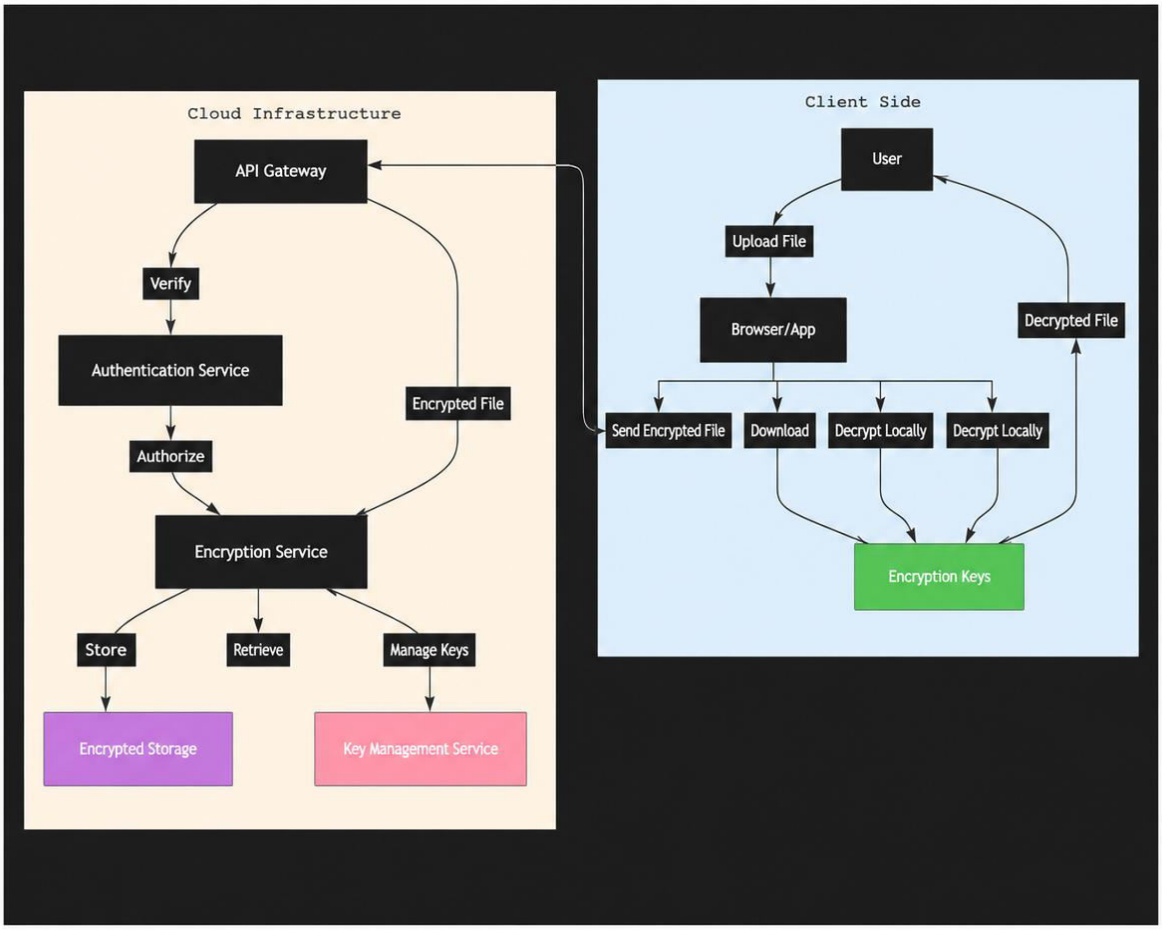
The proposed system introduces a secure cloud-based file encryption solution that ensures complete data confidentiality and user control before storing files in the cloud. In this system, all files are encrypted locally on the user's device using strong cryptographic algorithms such as AES for fast and secure file encryption and RSA for key protection. By implementing client-side encryption, the cloud provider never gains access to the original data or the encryption keys, eliminating the risk of unauthorized access from insiders or external attackers.

The system includes a secure authentication mechanism, user-friendly interface, and efficient key management process to generate, store, and retrieve encryption keys safely. Encrypted files are transmitted to the cloud over secure communication channels like SSL/TLS, ensuring protection against man-in-the-middle attacks. Only authorized users with valid decryption keys can retrieve and decrypt the files, guaranteeing end-to-end security.

IV. SYSTEM DESIGN AND ARCHITECTURE

The system architecture follows a client-server model designed for 'Zero-Knowledge' security. It includes a Client-Side Layer for local encryption/decryption and a Cloud Infrastructure Layer for backend services. The client-side handles user actions, local processing, and secure communication. The cloud infrastructure includes an API Gateway, Authentication Service, Encryption Service, Encrypted Storage, and Key Management Service (KMS). The system ensures that plaintext data never leaves the user's environment and is encrypted before transmission.

System architecture



V. SYSTEM IMPLEMENTATION

The system is implemented using a combination of technologies including Python 3.7.2, Flask, FastAPI, PyCryptodome, Cryptography.io, React, Web Crypto API, AWS S3, Google Cloud Storage, Firebase Storage, PostgreSQL, MongoDB, JWT, AES, RSA, PBKDF2, SSL/TLS, HTML5, and JavaScript. These technologies enable secure file encryption, user authentication, key management, and cloud storage integration.

The implementation follows a modular approach, with separate components for client-side encryption, server-side authentication, key management, and cloud storage. The client-side component handles file encryption and decryption using AES and RSA algorithms, while the server-side component manages user authentication, key storage, and cloud communication. The system integrates with cloud

storage platforms such as AWS S3, Google Cloud Storage, and Firebase Storage to provide scalable and reliable storage solutions.

VI. RESULTS AND DISCUSSION

System testing includes unit testing, integration testing, functional testing, performance testing, security testing, usability testing, and user acceptance testing. Test cases cover file encryption functionality, cloud upload, incorrect key decryption failure, authentication validation, and performance under load. The testing environment simulates real-world conditions with various devices and network conditions.

Results indicate reliable performance, accurate encryption/decryption, consistent cloud operations, strong security resistance, and user-friendly interface. The system demonstrates effective protection against unauthorized access and data breaches, ensuring the confidentiality and integrity of stored files. The testing process validates the system's ability to provide secure and efficient cloud storage services.

TABLE II. BLACK BOX TESTING TABLE

Table No.	CONTENT	Page No.
5 I	Block box testing table	43

TABLE III. TEST CASES

Table No.	CONTENT	Page No.
5.II	Test cases	45-46

VII. CHALLENGES AND LIMITATIONS

Several challenges were encountered during the development and implementation of the secure file storage system. Balancing strong client-side security with usability and performance, especially on resource-constrained devices, was a significant challenge. Enabling useful operations such as search, deduplication, and analytics without excessive leakage required careful consideration of cryptographic techniques and data handling methods.

Implementing practical, auditable key-management models that meet compliance requirements while keeping control with users posed additional challenges. Ensuring the quality of data for machine learning algorithms, managing time consumption for data acquisition, feature extraction, and retrieval, and addressing issues related to overfitting and underfitting were also significant concerns. The availability of expert resources and having clear objectives for business problems were additional challenges faced during the project.

VIII. CONCLUSION AND FUTURE SCOPE

The proposed secure file storage system effectively addresses the security and privacy concerns associated with cloud storage by implementing client-side encryption, strong cryptographic algorithms,

and role-based access control. The system ensures that plaintext data is never exposed during transmission or storage, providing end-to-end security for sensitive information. The integration of secure authentication, key management, and cloud storage platforms enables a robust and scalable solution for secure data management.

Future enhancements to the system may include integrating advanced cryptographic techniques such as homomorphic encryption or secure multiparty computation to enable secure computations on encrypted data. Incorporating machine learning-based anomaly detection for real-time threat identification, expanding multi-factor authentication and biometric verification, and supporting distributed cloud environments and multi-device synchronization are also potential areas for improvement. Adding features like encrypted search, automated key rotation, and tamper-proof audit logs using blockchain technology could further enhance the system's security and functionality.

Representative figures from the system are listed below:

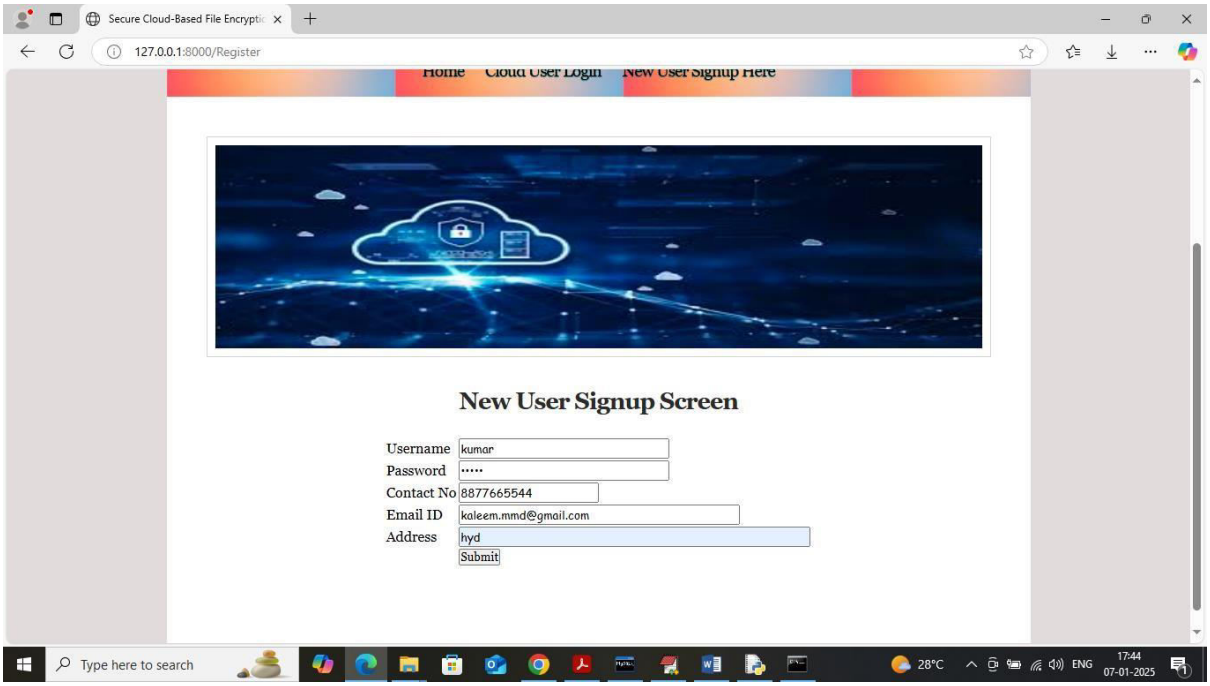


Fig. 1. Registration

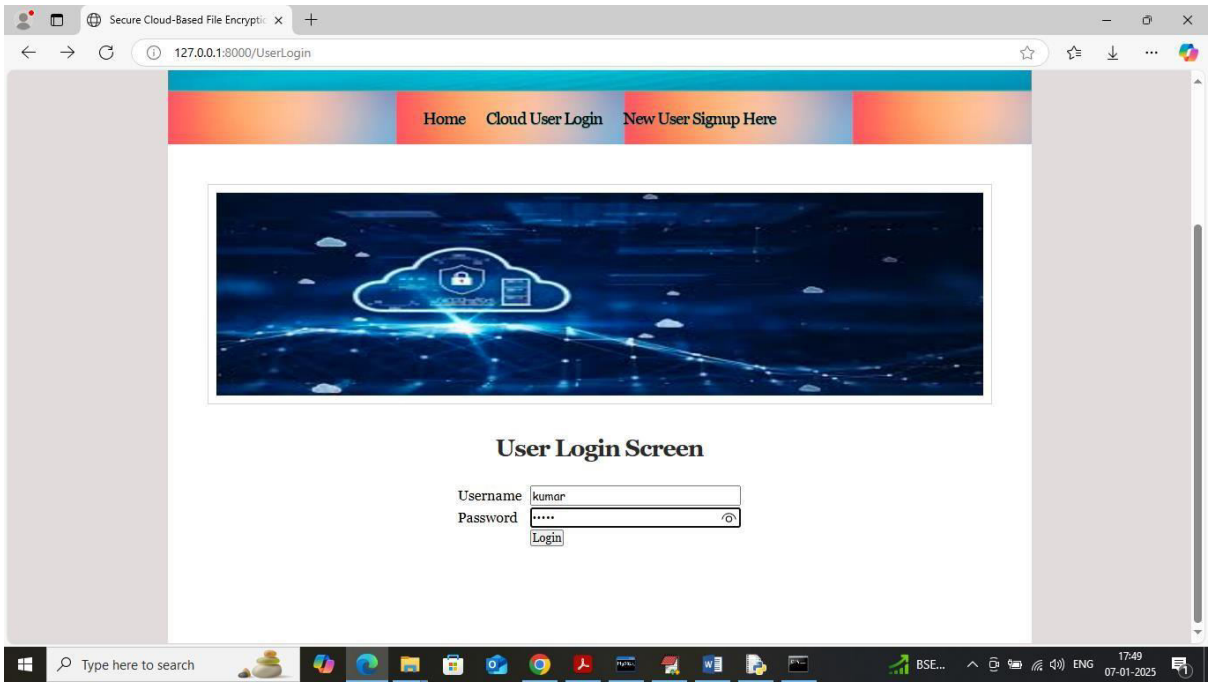


Fig. 2. Login Page

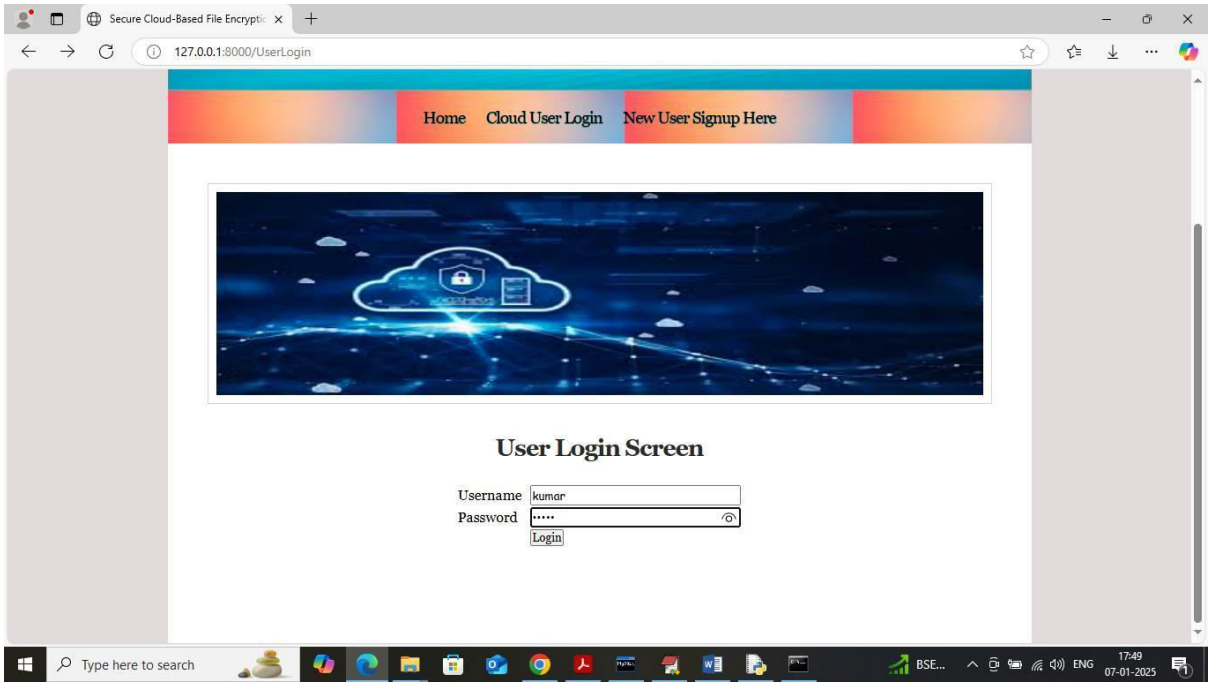


Fig. 3. Otp verification

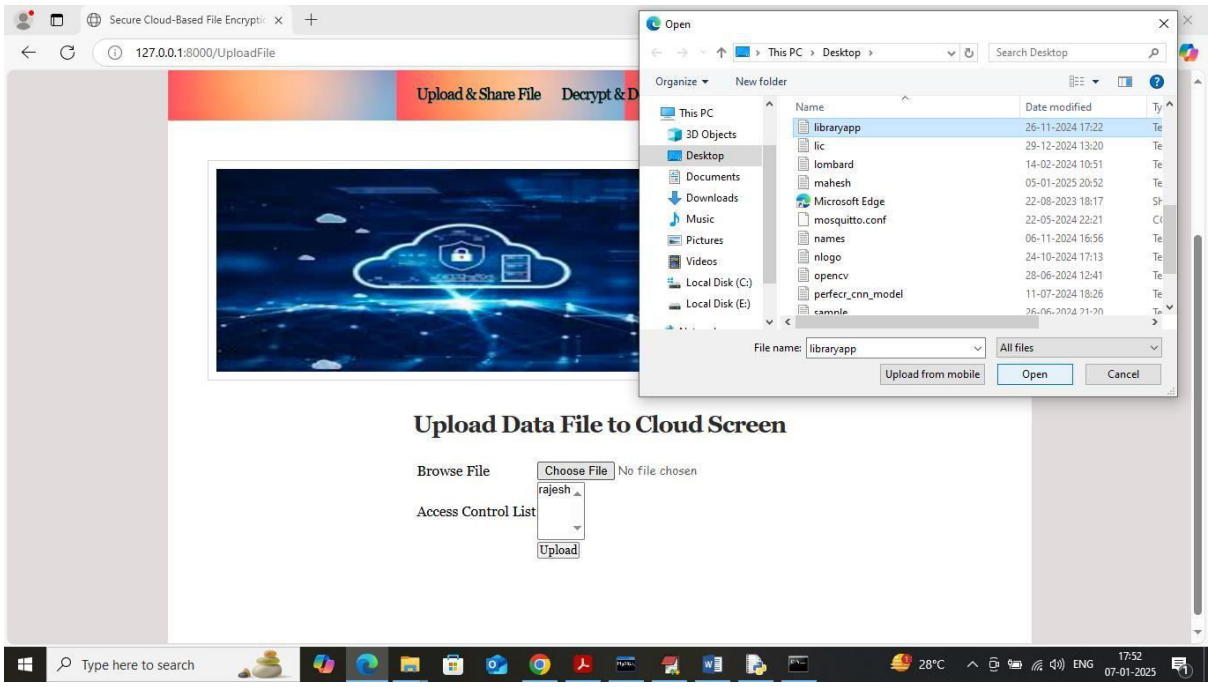


Fig. 4. Uploading the data File to Cloud

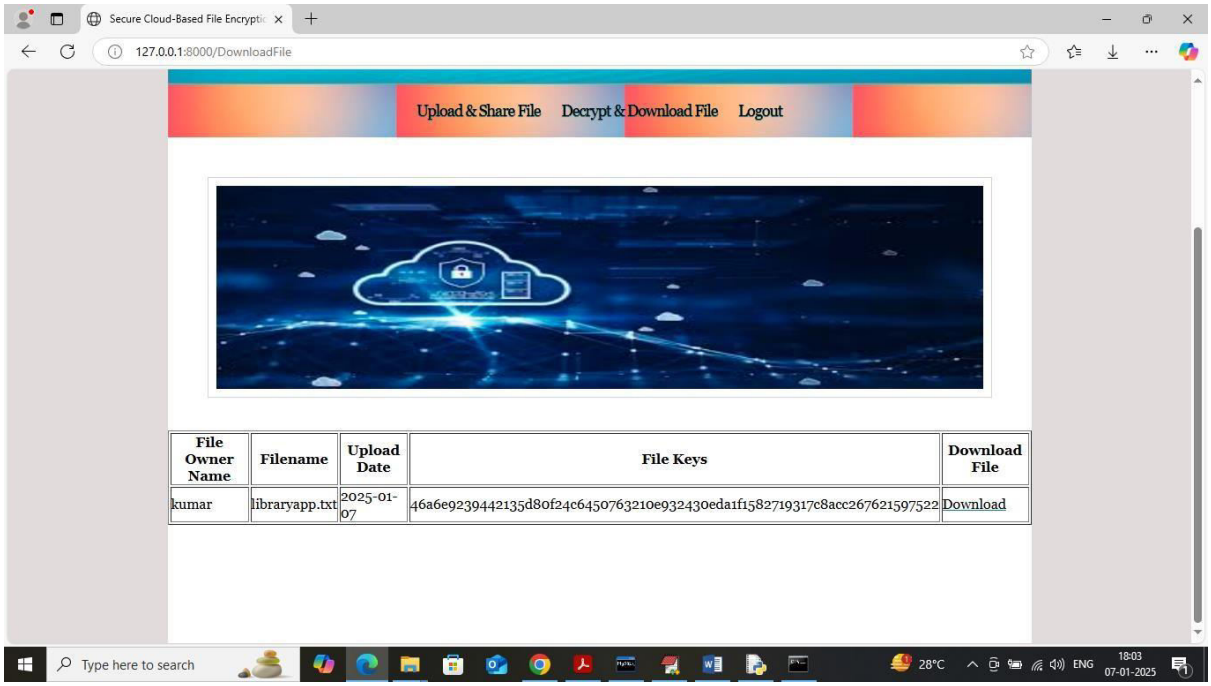


Fig. 5. Encryption

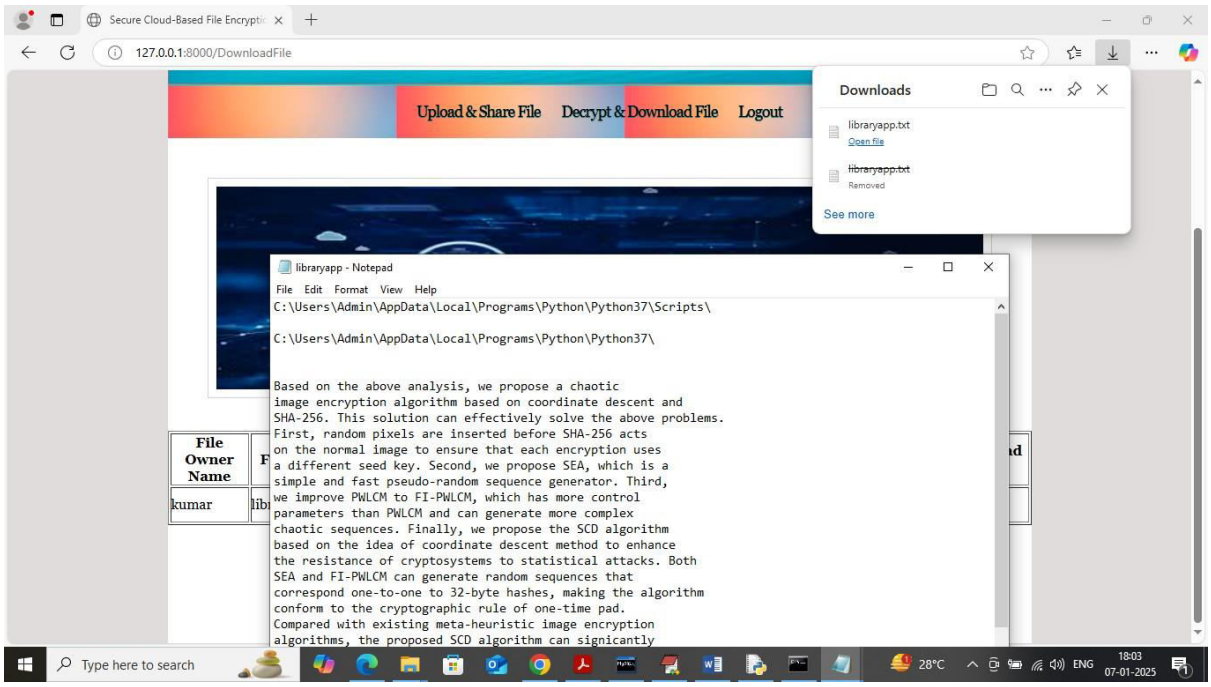


Fig. 6. Decryption

REFERENCES

[1] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[2] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, Jan.–Feb. 2012.

[3] M. Armbrust et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[4] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson, 2017.

[5] J. Li, M. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, pp. 131–143, Jan. 2013.

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology – EUROCRYPT 2005, LNCS vol. 3494, pp. 457–473, 2005.

[7] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," SIAM Journal of Computing, vol. 32, no. 3, pp. 586–615, 2003.

[8] K. Yang, X. Jia, K. Ren, and B. Zhang, "DAC-MACS: effective data access control for multiauthority cloud storage systems," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 121–133, June 2013.

[9] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Special Publication 800-145, 2011.

[10] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, June 2009.