Enhanced Cloud File Security Using Hybrid Cryptographic Algorithms Learning

GAGANSWAMY K S[1], JYOTHI K S[2]
[1,2] Channabasaveshwara Institute of Technology. Tumkur, Karnataka

Abstract- The rapid growth of digital data and cloud-based services has created new features in ensuring confidentiality, integrity, and secure access to sensitive information. Traditional single-algorithm encryption methods often struggle to balance strong security with efficiency, leaving systems vulnerable to evolving threats. This project presents a secure file storage framework that employs a hybrid cryptography approach, combining multiple symmetric encryption algorithms—AES, ChaCha20, and Fernet—in a rotating sequence. Uploaded files are divided into smaller chunks, each protected with a different algorithm, and the corresponding keys are bundled into an encrypted key blob secured by a master key. A minimal Flask-based backend is responsible for handling the processes of encryption, decryption, and file recovery, while the user interacts through a straightforward web interface for uploading and downloading files. This design strengthens security by avoiding reliance on a single algorithm, ensuring that the data remains safeguarded even if one technique is compromised. Practical testing indicates that the proposed approach achieves an effective balance between reliability, performance, and user convenience, making it suitable for secure file storage and accessing in today's computing environments.

Index Terms- Hybrid Cryptography for Secure Storage, AES Chacha20 and Fernet Encryption, Cloud-Based Confidential Data Protection, Encrypted Key Management Framework, Flask Backend for File Security.

I. INTRODUCTION

With the increasing adoption of cloud services by both organizations and individuals, safeguarding data and maintaining confidentiality have become critical challenges. This project focuses on addressing these challenges by designing and implementing a secure file storage framework that leverages cryptographic techniques to strengthen data protection.

Traditionally, cloud security solutions have relied on either symmetric or asymmetric cryptography, each offering unique benefits and drawbacks. Symmetric encryption is well known for its fast processing, making it effective for securing large amounts of data. However, it poses challenges in terms of securely sharing and managing the keys involved.

On the other hand, asymmetric encryption makes key exchange and authentication easier, but it requires greater computational effort, which can become inefficient when applied to very large datasets.

To overcome these limitations, the proposed system applies a hybrid cryptography strategy. By integrating the efficiency of symmetric algorithms for file encryption and the reliability of asymmetric algorithms for key management and authentication, the model achieves both security and performance. This two method ensures that sensitive files remain protected during storage and transfer in the cloud.

The overall aim of this project is to deliver a robust yet user-friendly solution for secure file storage. Beyond practical implementation, the work also provide to the growing field of data security by offering a balanced method that addresses the concerns of individuals and enterprises seeking to safeguard their information in cloud environments.

A. Problem Statement

The growth of cloud comput has changed data storage and access by providing scalability and ease of use. At the same time, this shift toward cloud-based services has heightened concerns about data confidentiality and privacy. Conventional encryption methods are often insufficient in addressing sophisticated and continuously evolving cyberattacks.

B. Related work

A number of authors have investigated hybrid cryptography to deal with the limitations of one-algorithm encryption in the cloud. A hybrid encryption model of cloud storage security was proposed by Bhatia and Singh [1], and it was established that a combination of a symmetric and asymmetric model can enhance the data confidentiality. Equally, Sahu et al. [2] have suggested an integrated cryptography technique to improve the security of files by balancing between speed and security.

The more recent researches are concerned with the more modern integrations. Indicatively, Sabo and Ismaila [3] introduced a secure file storage model based on hybrid cryptography targeted on the cloud platform. Khalaf and Sagheer [4] added blockchain to this method in order to bring transparency and confidence to hybrid encryption systems. Nandore and Kushwaha [5] have explored cloud-based approaches of sharing data securely based on hybrid approaches that focus on usability and the cryptographic strength.

In other works, algorithm-specific combinations are emphasized. AES and ECC protocols have been streamlined towards real-time protection by Verma and Dubba [7], and an ECC-AES hybrid model towards cloud security by Abualkas and Bhaskari [20]. Jagadeesh et al. [21] and Kumar and Kumar [22] examined AES-ECC variants which are modified to improve cloud file storage and report a better performance and resiliency.

Moreover, Patel and Thakkar [23] observed the hybrid cryptographic algorithms in order to increase the confidentiality of the cloud, and Sharma and Singh [24] described a hybrid cryptosystem to protect cloud files, which guarantees confidentiality, and integrity both. Research such as [25] and [29] by Prasad and Patel and Sahoo and Sahoo respectively contained optimization of hybrid AES-RSA designs towards file security, which provided faster processing and high encryption.

On the whole, the available studies invariably indicate that hybrid cryptography is a viable basis in achieving a reasonable performance-security tradeoff, which is why it may be an effective solution to secure cloud storage systems. Nevertheless, there is still a difficulty in streamlining the key management, computational cost, and scale of the system and this is the reason why the current work is on AES, ChaCha20, and Fernet-based hybrid storage safety.

II. COMPARISON WITH PREVIOUS WORK

Past studies in hybrid cryptography applied to cloud storage have been mostly dedicated to the hybridization of an asymmetric algorithm, such as RSA or ECC, with a symmetric algorithm AES with ECC or AES with RSA to trade off between speed and secure key transfer [1][20][25]. Although these methods enhanced confidentiality and integrity, they were usually characterized by high computing costs, single point of failure when it comes to algorithm dependency and intricate key management.

Others also implemented blockchain or Intel SGX enclaves to provide a higher level of trust and confidentiality [3][4], which added more complexity to the system and demanded specialized infrastructure. Other researches [22][23][29] maximised designs of hybrid AES-ECC or AES-RSA, although in this case, they still depended on one symmetric algorithm so in the event the selected cipher was broken, the whole system would be insecure.

The proposed project is ahead of these models that propose a multi-symmetric hybrid framework. This does not rely on a single cipher instead of rotating on AES, ChaCha20, and Fernet to make certain file chunks are encrypted using varied algorithms. Such diversification will contribute greatly to the risk of complete compromise in case one algorithm gets weak. In addition, keys are encrypted into an encrypted key blob which is protected by a master key and the keys are bundled together, which streamlines the process of key management and still offers high protection.

Moreover, the modern project does not need heavy infrastructure or special environments, as other previous designs did, but deploys its framework on a lightweight Flask backend with an elementary web interface. Such usability, performance, and layered security will make it stand out of the previous methods, as it is practically deployable, and resistant to changing threats.

III. METHODOLOGY

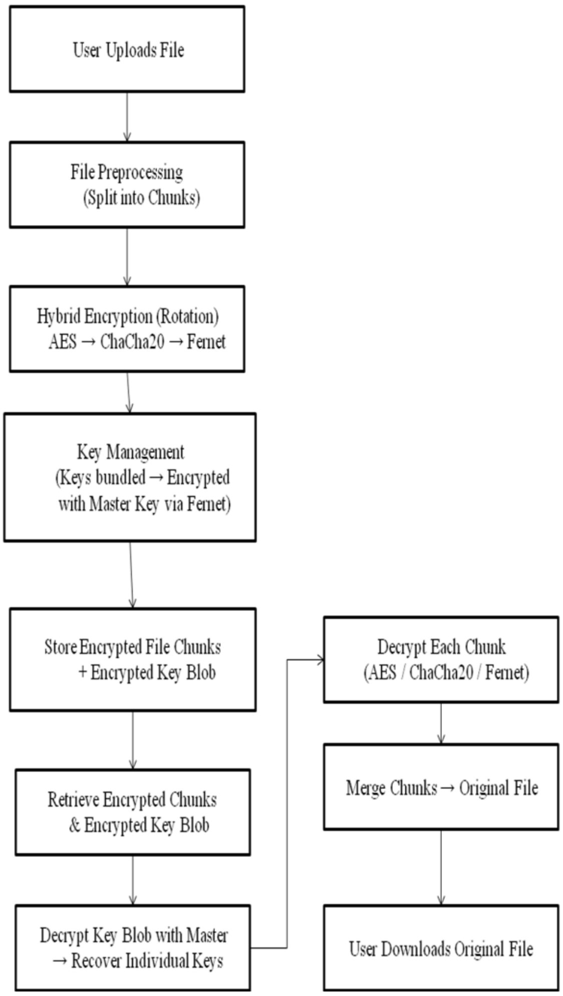

User Uploads File
File Preprocessing (Split into Chunks)
Hybrid Encryption (Rotation) AES → ChaCha20 → Fernet
Key Management (Keys bundled → Encrypted with Master Key via Fernet)
Store Encrypted File Chunks + Encrypted Key Blob
Retrieve Encrypted Chunks & Encrypted Key Blob
Decrypt Key Blob with Master → Recover Individual Keys
Decrypt Each Chunk (AES / ChaCha20 / Fernet)
Merge Chunks → Original File
User Downloads Original File

Fig 1: Methodology Flowchart

The methodology that will be used in this project is split into six major parts:

A. User Interaction

- Via a web interface developed with Flask, the user is allowed to upload a file.
- Text / binary The file is received and forwarded to the backend.

B. File Preprocessing

- The file that is uploaded is broken up into smaller bits.
- This is to guarantee parallel encryption and minimize the chances of mass encroachment.

C. Hybrid Encryption Process

- The files are encrypted using a varying symmetric algorithm in a rotating order (AES, ChaCha20, Fernet).
- This will prevent reliance on one algorithm.
- For example:
- Chunk 1 → AES
- Chunk 2 → ChaCha20
- Chunk 3 → Fernet
- Chunk 4 AES (rotation persists).

D. Key Management

- The secret key is generated by each of the algorithms.
- A key blob is encrypted with Fernet using a master key with all these keys.
- The master key is kept safe and availed to the user to be decrypted.

E. File Storage and Retrieval

- The ciphertext blocks and the ciphertext key block are maintained on the system/cloud.
- In case a user asks to have a download, the system will get the encrypted chunks and the key blob.

F. File Recovery and Decryption.

- The encrypted key blob is unlocked using the master key.
- The algorithm is used to decode each fragment of file.

All the decrypts are repackaged in the original file and availed as a download.be written and together compiled to form a complete research ready for Peer review.

IV. RESULTS

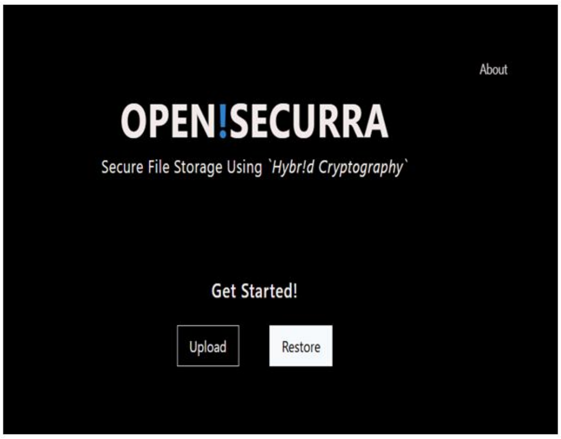
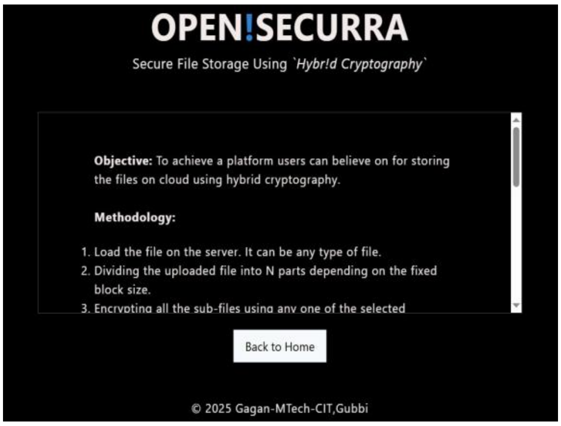


Fig. 2: Main page (when our code is compiled)



OPEN!SECURRA

Secure File Storage Using `*Hybr!d Cryptography*`

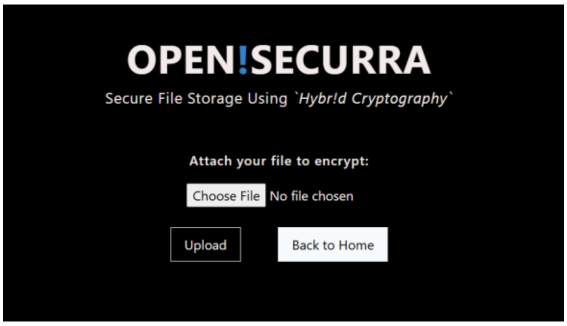
Objective: To achieve a platform users can believe on for storing the files on cloud using hybrid cryptography.

Methodology:

1. Load the file on the server. It can be any type of file.
2. Dividing the uploaded file into N parts depending on the fixed block size.
3. Encrypting all the sub-files using any one of the selected

© 2025 Gagan-MTech-CIT,Gubbi

Fig 3: About page

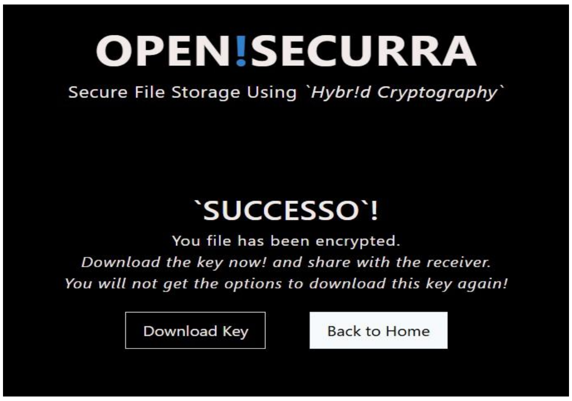


OPEN!SECURRA

Secure File Storage Using `*Hybr!d Cryptography*`

Attach your file to encrypt:

Fig. 4: To encrypt file



OPEN!SECURRA

Secure File Storage Using `*Hybr!d Cryptography*`

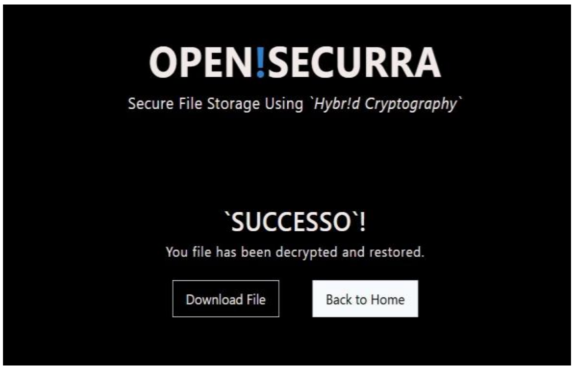
`SUCCESSO`!

You file has been encrypted.

Download the key now! and share with the receiver.

You will not get the options to download this key again!

Fig. 5: Encrypted key



OPEN!SECURRA

Secure File Storage Using `*Hybr!d Cryptography*`

`SUCCESSO`!

You file has been decrypted and restored.

Fig. 6: To download the original file

CONCLUSION

The key components of the user-friendly interface for uploading files and utilizing the advanced encryption methods make the cloud-based secure file storage system more efficient in terms of data security. Through this integration, system allows users to upload files of various formats and data types. This makes the system more accessible and convenient, and at the same time, it has stringent security measures. After uploading file, they get encrypted using the advanced algorithms that ensure the security of your data and protection from cyber threats.

Users are also given the ability to download a security key which can help safeguard the privacy and security of their data. With that key users may rest assure that their files remain impenetrable because no record of the original file or the key is kept by the system. These strategies by not only ensuring security of data but also give power to the user by allowing them to manage their digital information with confidence and satisfaction.

In addition to its security features, the system simplifies the process of uploading and downloading files, ensuring that encrypted data can be restored to its original form with less effort. This gives users confidence that their files remain protected throughout the encryption and decryption process, with no risk of external interference. By prioritizing user involvement in data protection, the system strengthens trust and demonstrates its commitment to safeguarding digital information.

Overall, the combination of a user-friendly interface for file handling, along with effective key management, represents a significant advancement in cloud security practices. By merging ease of use with strong encryption techniques, the system empowers users to preserving user control of their datadata without sacrificing convenience. Such innovations play a vital role in encouraging both individuals and organizations to adopt cloud services securely and confidently.

REFERENCES

[1] A. Bhatia and R. Singh, *"Hybrid encryption model for cloud storage security," Int. J. Comput. Appl.*, vol. 179, no. 24, pp. 1–6, 2018.

[2] S. K. Sahu, R. K. Gupta, and P. K. Pattnaik, *"A combined cryptography method for secure cloud storage," Procedia Comput. Sci.*, vol. 92, pp. 339–346, 2016.

[3] V. Z. Sabo and J. M. Ismaila, *"Secure cloud file storage using hybrid cryptography," Int. J. Emerg. Multidiscip.: Comput. Sci. & AI*, vol. 4, no. 1, pp. 24–31, 2025.

[4] F. M. Khalaf and A. M. Sagheer, *"Blockchain-enabled hybrid encryption for cloud data storage," J. Inf. Syst. Eng. & Manag.*, vol. 10, no. 22s, 2025.

[5] T. Nandore and S. Kushwaha, *"Hybrid encryption method for cloud data sharing," Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 12, no. 7, pp. 123–130, 2023.

[6] A. S. Kumar et al., *"Cloud file security using hybrid cryptographic methods," Int. J. Adv. Res.*, vol. 11, pp. 1–5, Apr. 2023.

[7] H. Verma and N. M. Dubba, *"Enhancement of AES and ECC protocols for real-time data protection," Metall. Mater. Eng.*, vol. 31, no. 5, pp. 37–46, May 2025.

[8] R. K. Bairwa et al., *"A hybrid cryptographic framework for cloud file security," Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 4, pp. 300–306, 2023.

[9] M. Joshi and M. Prateek, *"Hybrid algorithm performance study for cloud data security," J. Digit. Economy*, vol. 3, no. 2, 2024.

[10] L. Archana et al., *"Data storage security using hybrid techniques in cloud," Int. J. Eng. Technol.*, vol. 7, no. 2.20, pp. 150–152, 2018.

[11] A. Chaudhari and P. R. Bhaladhare, *"Survey on hybrid cryptography in cloud file protection," Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, 2023.

[12] S. M. K. Wani and A. Kumar, *"Secure file storage with hybrid cryptographic design," Int. J. Res. Eng. Sci. Manag.*, vol. 5, no. 5, pp. 35–39, 2022.

[13] N. Kumari and V. Malhotra, *"Hybrid cryptography for cloud data security," IJRASET J. Appl. Sci. Eng. Technol.*, vol. 10, no. 8, pp. 1256–1263, 2022.

[14] M. Bhawane and S. Ahuja, *"Hybrid cryptography-based secure data storage in cloud," IJRASET*, vol. 12, no. 2, pp. 800–805, 2024.

[15] K. Reddy et al., *"File storage security using hybrid cryptography with fragmentation," J. Survey Fish. Sci.*, vol. 10, no. 1, 2023.

[16] A. S. Ghadi, *"Secure file storage in cloud using hybrid cryptographic system," Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 12, pp. 1200–1207, Dec. 2020.

[17] S. Sankar et al., *"Hybrid file-sharing security using encryption and network theory,"* in *Data Sci. & Appl. (ICDSA)*, LNNS, vol. 1237, Springer, pp. 141–150, 2025.

[18] J.-F. Lai and S.-H. Heng, *"Hybrid cryptography approach for secure cloud file storage," J. Informatics & Web Eng.*, voll. 1, no. 2, pp. 1–18, 2022.

[19] N. Ghule et al., *"Content-aware hybrid cryptography for securing file storage in cloud environment,"* in *Int. Conf. Commun. & Artif. Intell.*, Springer, pp. 371–383, 2021.

[20] Y. M. A. Abualkas and D. L. Bhaskari, *"ECC-AES hybrid model for cloud security," Int. J. Eng. Trends & Technol.*, vol. 72, no. 4, pp. 92–100, 2024.

[21] S. Jagadeesh et al., *"AES-modified ECC hybrid encryption for secure cloud storage," J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 30, no. 1, pp. 230–240, 2024.

[22] S. Kumar and D. Kumar, *"Cloud data protection using AES-ECC hybrid approach," J. Mobile Multimedia*, vol. 17, no. 3, pp. 531–546, 2021.

[23] R. Patel and V. Thakkar, *"Cloud confidentiality via hybrid cryptographic algorithms," Int. J. Comput. Appl.*, vol. 176, no. 9, pp. 22–27, 2019.

[24] A. Sharma and P. Singh, *"Hybrid cryptosystem for cloud-based file protection," Int. J. Eng. Res. Technol.*, vol. 8, no. 6, pp. 134–140, 2020.

[25] B. R. Prasad and K. K. Patel, *"AES and RSA based hybrid file security in cloud," Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 45–50, 2020.

[26] C. Wang, Q. Wang, K. Ren, and W. Lou, *"Data storage protection in cloud computing," IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 587–600, 2011.

[27] M. Storch and C. A. F. de Rose, *"Cost modeling for cryptographic cloud file systems,"* in *Proc. 25th Euromicro Int. Conf. Parallel, Distrib. & Netw.-Based Process. (PDP)*, pp. 9–14, 2017.

[28] V. Masthanamma and G. L. Preya, *"RSA-based encryption for cloud security," Int. J.*

Innov. Res. Sci. Eng. Technol., vol. 4, no. 3, pp. 1441–1445, 2015.

[29] P. K. Sahoo and S. R. Sahoo, *"Optimized hybrid encryption for secure cloud file storage," Int. J. Cloud Comput.*, vol. 9, no. 4, pp. 345–360, 2020.

[30] D. Bansal and R. K. Gupta, *"Hybrid security design for cloud storage with cryptography and access control," Int. J. Netw. Secur. & Appl.*, vol. 12, no. 2, pp. 15–24, 2020.