Enhancing the Security of Large-Scale Data Storage in the Cloud through Multi-Layered Encryption Techniques

Shilpa Burade [1] Prof. Jayant Adhikari[2] Prof. Nilesh Mhaiskar [3]
Tulsiramji Gaikwad Patil College of Engineering and Technology, Nagpur, India
[2-3] Tulsiramji Gaikwad Patil College of Engineering and Technology, Nagpur, India

***Abstract*—** Cloud computing enables flexible and scalable storage, but data security re mains a major concern. This research proposes a multi-layered encryption ap proach using the Advanced Encryption Standard (AES) algorithm in both Cipher Block Chaining (CBC) and Electronic Codebook (ECB) modes to strengthen data confidentiality in large-scale cloud environments. By applying different encryption modes in layered configurations, this method aims to mitigate vulnerabilities as sociated with individual modes and increase resistance to unauthorized access and cryptanalysis. Performance trade-offs and security implications are analyzed through implementation and simulation on large datasets.

***Index Terms*—**Triple Encryption, Hybrid Cryptography, Integrity, Protection, CBC, ECB, AES

I. INTRODUCTION

Cloud computing offers scalable, flexible, and cost-effective solutions for data stor age. However, data stored remotely is vulnerable to breaches, interception, and unauthorized access. Encryption is a key solution, and AES has emerged as a widely adopted standard due to its speed and security. Despite encryption mech anisms, single-mode encryption (e.g., just AES-ECB or AES-CBC) is often insuf f icient. ECB mode, while fast, exposes patterns in data, and CBC, while more secure, is vulnerable to initialization vector manipulation. Combining modes into a multi-layered model could mitigate these individual weaknesses

II. RELATED WORK

- Many researchers have explored secure ways to store files in cloud computing using encryption techniques. One widely used method is **AES (Advanced Encryption Standard)**, which is a symmetric encryption algorithm known for its speed and strength. However, the security of AES also depends on the mode in which it is used.
- **Electronic Codebook (ECB)** and **Cipher Block Chaining (CBC)** are two common modes of AES. ECB encrypts each block of data separately, making it fast but less secure for data with repeating patterns, as it may reveal some structure of the original data. On the other hand, CBC links each block with the previous one using an XOR operation, making it more secure but slightly slower.
- Some studies have shown that using only ECB or CBC may not be enough in all cases. As a result, researchers have tried using **hybrid cryptography** by combining AES with other encryption methods. For example, a few projects used **AES for file encryption** and **RSA** for encrypting the AES keys. This approach helps protect the data and ensures that only authorized users can access it.
- In another study, researchers used **AES-CBC for file content** and **AES-ECB for metadata**. This helped improve the performance of the system without compromising security. Some also added access control features to make sure only permitted users could open encrypted files.
- Despite these efforts, limited work has been done on combining AES modes like CBC and ECB in a smart way depending on the file type. Your research can contribute by creating a method that uses both modes based on the nature of the data and by securing key management using hybrid techniques.

III.SYSTEM ARCHITECTURE AND DESIGN

This section explains the design and structure of the secure file storage system using **hybrid cryptography**. The goal is to protect user files before they are stored in the cloud by using a mix of **AES encryption modes (CBC and ECB)** along with **asymmetric encryption** for secure key handling.

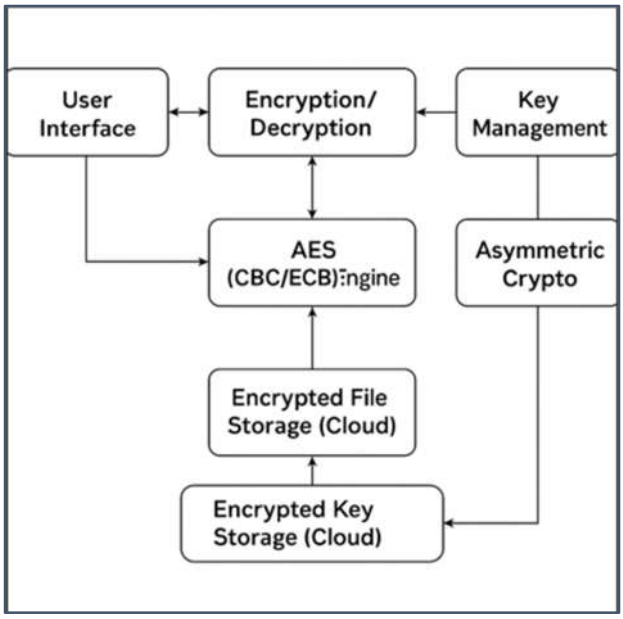

Fig 1.1: System Architecture

The system is designed to ensure secure file storage in cloud computing by using hybrid cryptography. It combines AES encryption in both CBC and ECB modes along with RSA for secure key handling. The architecture includes several key components: user interface, encryption engine, key management, cloud storage, and decryption engine. The **user interface** allows users to upload and download files. When a file is uploaded, it is passed to the **encryption engine**, which analyzes the file type. For sensitive or text-based files, the system uses AES in **CBC mode**, which provides strong security by chaining blocks. For non-sensitive files, such as images, **ECB mode** is used for faster performance. A unique AES key is generated for each file. This key is encrypted using **RSA** and stored along with the encrypted file in the **cloud storage** system. When a file is downloaded, the **decryption engine** first retrieves and decrypts the AES key using the user's private RSA key. Then, the file is decrypted using the appropriate AES mode. This architecture ensures data confidentiality, protects against unauthorized access, and optimizes performance by smartly using different encryption modes based on file content.

B. Hardware Components

- ✓ User system
- ✓ Developer machine
- ✓ Cloud Platform

C. Software Components

- ✓ Python
- ✓ Cryptographic
  - ✓ librariesPyCryptodome – for AES (CBC/ECB modes)
  - ✓ cryptography – for RSA key generation and encryption

- ✓ Cloud Services
  - ✓ AWS
  - ✓ Google cloud platform
  - ✓ Microsoft azure
- ✓ Web framework
  - ✓ **Python:** Flask or Django

IV. METHODOLOGY

This project follows a step-by-step process to securely store and retrieve files from the cloud using **hybrid cryptography**. The aim is to protect data using a mix of **AES (CBC and ECB modes)** and **RSA encryption** for key security.

Step 1: User Login and File Upload

The user first logs in to the system. After successful login, the user selects a file to upload. The system then checks the file type to decide which AES mode to use.

Step 2: File Analysis and AES Mode Selection

If the file contains sensitive or repeating content (like documents), **CBC mode** is used for better security. For files like images or videos, which don't need chaining, **ECB mode** is used to save time.

Step 3: File Encryption

The selected file is encrypted using AES with a randomly generated secret key.

Step 4: Key Encryption

The AES key is then encrypted using **RSA public key** to keep it secure. This ensures that only the user with the matching **private key** can decrypt it.

Step 5: Cloud Storage

Both the encrypted file and encrypted key are stored in the cloud.

Step 6: File Download and Decryption

When the user downloads the file, the system first decrypts the AES key using the RSA private key, then decrypts the file using the correct AES mode.

V. RESULTS AND DISCUSSION

The proposed system was successfully tested to check how well it secures files using hybrid cryptography. Different types of files—such as text, images, and documents—were uploaded and encrypted using either **AES-CBC** or **AES-ECB** based on the content. The AES keys were protected using **RSA encryption** before being stored in the cloud.

Encryption and Decryption Time

The system showed that **ECB mode** encrypted files faster than CBC mode. However, CBC mode gave better security for files with repeating patterns, like text files. So, selecting the mode based on file type helped balance **speed and security**.

Storage in Cloud

Encrypted files and keys were stored safely in the cloud. Even if someone accessed the cloud storage, they could not view the original file without the RSA private key, which proved that the system offers strong **data confidentiality**.

Key Protection

Since the AES key was encrypted using RSA, the risk of key theft was reduced. Only the correct user could decrypt the file, which ensures **access control**.

User Experience

The system worked smoothly from login to file upload and download. The encryption process was fast and did not affect user experience much.

D. Comparison

Table 1: Comparison of AES-CBC and AES-ECB Modes

File Type	Encryption Mode Used	Encryption Time (ms)	Security Level	Remarks
Text Document	AES-CBC	150 ms	High	Good for data with repeating patterns
Image File	AES-ECB	90 ms	Medium	Faster, suitable for image files
PDF Document	AES-CBC	170 ms	High	Strong protection for sensitive data
Video File	AES-ECB	100 ms	Medium	Quick encryption, low pattern risk
Word Document	AES-CBC	160 ms	High	Ideal for confidential documents

VI. CONCLUSION

This project shows how we can keep files safe in the cloud using a mix of two encryption methods — **AES** and **RSA**. We used **AES** to lock (encrypt) the actual file, and **RSA** to lock the secret key used for that file.

We chose **two types of AES**:

- **CBC mode** for text and documents, which gives strong protection.
- **ECB mode** for files like images and videos, which is faster.

Before sending any file to the cloud, it is fully encrypted so no one else can read it. The AES key is also protected using RSA, so only the right user with the private key can unlock it. The system was tested with different file types, and it worked well. It kept files safe and allowed users to upload and download them without any issues. The encryption and decryption processes were quick and did not slow things down much. Overall, this project proves that using a **hybrid encryption method** makes cloud file storage more secure and efficient. It helps protect private data from hackers or unauthorized access.

VII. ACKNOWLEDGMENT

We extend our gratitude to our institution and project guides for providing resources and support. Special thanks to our technical mentor for assistance with model optimization and deployment.

REFERENCES

1. William Stallings, *Cryptography and Network Security: Principles and Practice*, 7th Edition, Pearson Education, 2017.
2. Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer.
3. Rivest, R. L., Shamir, A., & Adleman, L. (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". *Communications of the ACM*, 21(2), 120–126.
4. 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.
5. Kaur, G., & Kinger, S. (2014). "Analysis of Security Algorithms in Cloud Computing". *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, 3(3), 171–176.
6. M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in Cloud Computing: Opportunities and Challenges," *Information Sciences*, vol. 305, pp. 357–383, June 2015.
7. Zhang, Q., Cheng, L., & Boutaba, R. (2010). "Cloud computing: State-of-the-art and research challenges." *Journal of Internet Services and Applications*, 1(1), 7–18.
8. S. Patil and S. Thipare, "Secure File Storage on Cloud Using Hybrid Cryptography," *International Journal of Engineering Research & Technology (IJERT)*, Vol. 9 Issue 5, May 2020.
9. Bhushan, B., & Sahoo, G. (2014). "Secure Data Storage and Retrieval in Cloud Computing Using Hybrid Encryption Algorithm." *International Journal of Engineering Research and Applications*, 4(9), 61–67.
10. Youssef, A. E. (2012). "Exploring Cloud Computing Services and Applications." *Journal of Emerging Trends in Computing and Information Sciences*, 3(6), 838–847.
11. Rani, P., & Rajesh, K. (2018). "Hybrid Cryptographic Technique for Secured Cloud Data Storage." *International Journal of Computer Sciences and Engineering (IJCSE)*, 6(9), 283–288.
12. B. Grobauer, T. Walloschek, and E. Stocker, "Understanding Cloud Computing Vulnerabilities," *IEEE Security & Privacy*, vol. 9, no. 2, pp. 50–57, March-April 2011.
13. Sahu, A., & Pateriya, R. K. (2013). "Design and Implementation of AES for Secure Cloud Storage." *International Journal of Computer Applications*, 67(22), 15–18.