JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA SECTOR-62

MINOR PROJECT SYNOPSIS
ODD SEMESTER 2025

PROJECT TITLE: SECURE CLOUD STORAGE USING DYNAMIC AES, BLOCKCHAIN, AND ECC

Faculty Coordinator: Dr. Aastha Maheshwari

SUBMITTED BY:
ARNAV SINGH (23103369)
DAKSH (23103136)
VANSH RATHI (23103134)

Panel Teachers: Dr. Ritika Chaudhary , Dr. Silki Kharaliya

Abstract

With the rapid growth of cloud computing, securing sensitive data stored in cloud environments has become a critical challenge. Traditional encryption techniques often rely on static keys, which are prone to compromise and make systems vulnerable. This project proposes a novel solution by integrating **Dynamic AES (Advanced Encryption Standard)** for file-level encryption, **Blockchain** for tamper-proof and decentralized key management, and **ECC (Elliptic Curve Cryptography)** for secure key exchange. Together, these techniques provide confidentiality, integrity, and efficient key distribution in cloud storage. The proposed system ensures that each file is encrypted with a unique dynamic AES key, stored securely in a blockchain mini-ledger, and shared using ECC. This model enhances data security, reduces risks of unauthorized access, and ensures trust in cloud-based storage systems.

Introduction

Cloud storage offers scalability and cost-effectiveness but raises serious concerns about **data breaches, privacy violations, and weak key management**. Traditional methods rely heavily on central authorities or static encryption keys, which are vulnerable to attacks. To overcome these limitations, hybrid models combining symmetric and asymmetric encryption with blockchain-based integrity are gaining attention. This project leverages **dynamic AES encryption** for stronger confidentiality, **blockchain** for immutable logging of encryption keys, and **ECC** for lightweight, secure key sharing.

Problem Statement

The primary challenge in cloud security is balancing **strong encryption with efficient performance**. Existing solutions face the following issues:

- Static AES keys → higher risk of brute force or replay attacks.
- Centralized key management → single point of failure.
- RSA-based key exchange → computationally heavy and inefficient for real-time systems.

Hence, a scalable and efficient model that integrates **dynamic AES, blockchain, and ECC** is required to ensure **data confidentiality, tamper-proof integrity, and secure sharing** in cloud storage systems.

Literature Review

- The first reference study introduced **Dynamic AES with Blockchain-based key management**, proving that combining tamper-proof ledgers with encryption enhances trust and auditability in cloud storage .
- Another study proposed **SymECCipher**, a hybrid **ECC–AES model**, showing significant improvements in encryption speed (5ms vs RSA's 15ms) and decryption efficiency, making it suitable for real-time cloud applications s41598-025-01315-5
  .
- Both works highlight that **hybrid cryptographic models** combining symmetric (AES) and asymmetric (ECC) methods with blockchain improve scalability, reduce overhead, and provide **quantum-resistant security** for future applications.

Key Features

1. **Dynamic AES Encryption** – Each file is encrypted with a unique, one-time AES key.

2. **Blockchain Mini-Ledger** – Keys are stored in a tamper-proof chain, ensuring immutability and traceability.

3. **ECC for Secure Key Sharing** – Provides lightweight, high-security public–private key exchange.

4. **User-Friendly APIs** – FastAPI-based REST endpoints for upload, download, and share operations.

5. **Lightweight Implementation** – SQLite for local metadata storage, Docker for deployment.

Methodology

1. **Encryption Module**: When a user uploads a file, it is encrypted with a dynamic AES key.

2. **Blockchain Module**: The AES key is stored in a blockchain mini-ledger with timestamp and hash references.

3. **ECC Key Exchange**: For file sharing, the AES key is encrypted using the recipient's ECC public key.

4. **Decryption Module**: Recipient decrypts the AES key with their ECC private key and retrieves the file.

5. **Testing & Validation**: Verify immutability of blockchain records, encryption speed, and successful file sharing.

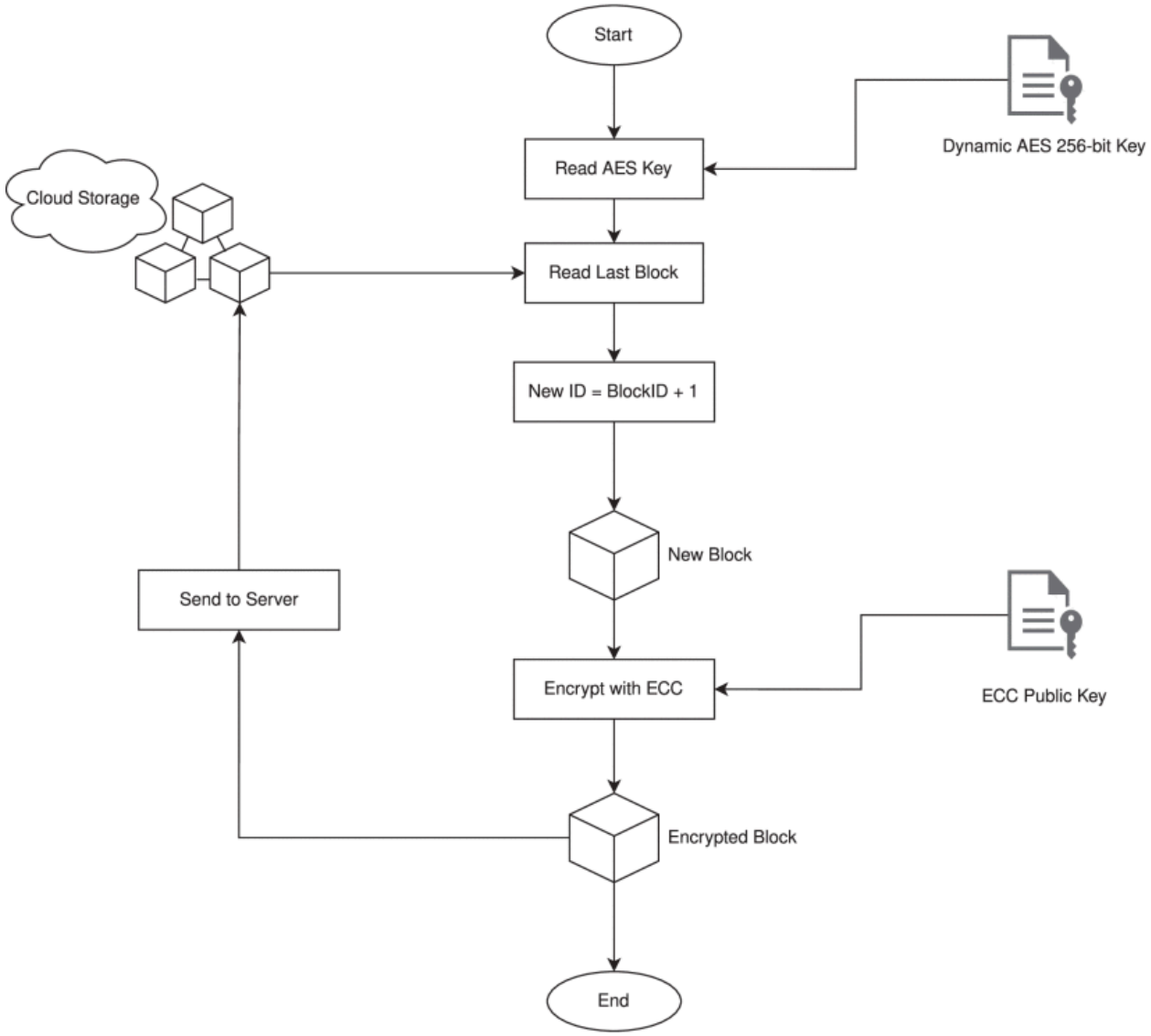

Figure 1: Workflow of Secure Cloud Storage Using Dynamic AES, Blockchain, and ECC

Technologies Used

- **Programming Language**: Python
- **Framework**: FastAPI (for REST APIs)
- **Cryptography Libraries**: PyCryptodome (AES), Cryptography (ECC)
- **Blockchain**: Custom Python-based mini-ledger
- **Database**: SQLite
- **Deployment Tools**: Docker, GitHub

Benefits

- Ensures **end-to-end data confidentiality** in cloud storage.
- Prevents **tampering or deletion of keys** with blockchain immutability.
- ECC provides **fast, lightweight, and secure sharing** compared to RSA.
- Scalable and cost-effective for academic and enterprise use.
- Can serve as a **foundation for secure file-sharing platforms**.

Future Scope

- Extend blockchain to a **distributed peer-to-peer network** for higher trust.
- Implement **quantum-resistant cryptography** (lattice-based models).
- Integrate **multi-factor authentication** for user-level security.
- Deploy on real cloud platforms like **AWS or Azure**.
- Apply to sensitive domains like **healthcare, banking, and government data storage**.

Conclusion

This project presents a **hybrid secure cloud storage model** by integrating **Dynamic AES, Blockchain, and ECC**. It resolves the limitations of traditional encryption by ensuring **unique per-file keys, tamper-proof storage, and lightweight secure sharing**. The implementation demonstrates improved **security, performance, and scalability**, making it suitable for academic research as well as real-world applications.

References

1. Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security. (2023).

2. Selvi, P., & Sakthivel, S. (2025). *A Hybrid ECC-AES Encryption Framework for Secure and Efficient Cloud-Based Data Protection*. Scientific Reports. https://doi.org/10.1038/s41598-025-01315-5