International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Owner Bot: Secure Cloud Data Management Using Blockchain and Homomorphic Encryption

Authors: Dr. K. Anandan
Assistant Professor, Department of Computer Applications, Nehru College of Management, Coimbatore

V. Dinesh Kumar
II MCA, Department of Computer Applications,
Nehru College of Management, Coimbatore, Tamil Nadu, India

## ABSTRACT

Cloud storage services have gained significant popularity due to their fundamental support for the rapid growth of cloud computing. However, despite their advantages, security risks such as data leakage, mismanagement, and malicious attacks still persist. This project proposes a security model that combines cloud computing with blockchain technology to ensure data integrity while maintaining confidentiality through homomorphic encryption.

## 1. INTRODUCTION

With the widespread adoption of cloud computing, there has been an exponential increase in the volume of data being stored and processed by cloud service providers. While cloud storage offers scalability, flexibility, and cost-efficiency, it also introduces significant security risks, including data breaches and unauthorized access. This paper presents a framework integrating homomorphic encryption and blockchain technology to secure cloud data.

## 2. LITERATURE REVIEW

Several studies have been conducted on cloud security:

- **Singh & Singh (2020)** surveyed cloud computing models, security, and privacy issues.
- **Zhang & Liu (2019)** analyzed cloud storage security challenges and solutions.
- **Li et al. (2017)** explored data security in cloud computing, emphasizing encryption techniques.
- **Zhao & Li (2018)** proposed blockchain-based data security models.
- **Patel & Mishra (2019)** examined cloud storage security techniques and privacy models.

## 3. SECURITY MODEL APPROACHES

- ### Homomorphic Encryption

  Homomorphic encryption allows computations to be performed on encrypted data without requiring decryption. This ensures privacy while enabling secure processing.

- ### Blockchain Integration

  Blockchain technology ensures data integrity by creating a decentralized, tamper-proof ledger of all data transactions.

- ### Secure Key Management

  A cryptographic key management system ensures only authorized users can access data through public-private key pairs.

## 4. METHODOLOGY

- **Data Encryption and Storage**

Sensitive data is encrypted before being uploaded to the cloud. A blockchain ledger records all encryption and decryption transactions.

- **Data Sharing and Access Control**

Access requests are logged on the blockchain. The Owner Bot Model facilitates secure data exchange between data owners and authorized users.

- **Audit and Monitoring**

A blockchain-based audit system enables data owners to track and verify data access and modifications.

## 5. EXISTING SYSTEM

The traditional cloud security methods rely on:

- Standard encryption techniques (AES, RSA) that require decryption for processing.
- Centralized security models prone to unauthorized access and data breaches.
- Limited tracking and auditing mechanisms, leading to potential tampering.
- Weak authentication mechanisms, increasing the risk of cyberattacks.

## 6. PROPOSED SYSTEM

The proposed Owner Bot Model introduces:

- Homomorphic encryption to enable encrypted data processing without decryption.
- Blockchain-based verification to prevent unauthorized modifications.Secure key management to ensure controlled access to data.
- Decentralized audit logs for transparent data tracking and monitoring.

## 7. ADVANTAGES

- Ensures sensitive data remains encrypted during processing.
- Enhances integrity verification using blockchain.
- Provides tamper-proof audit logs for accountability.
- Supports scalable and efficient cloud security solutions.
- Automates secure data sharing between users and owners.

## 8. CODE IMPLEMENTATION (EXCERPT)

```
from cryptography.fernet import Fernet import
hashlib


def generate_key():
    return Fernet.generate_key()


def encrypt_data(data, key): cipher =
    Fernet(key)
    return cipher.encrypt(data.encode())


def decrypt_data(encrypted_data, key): cipher =
    Fernet(key)
```

```
    return cipher.decrypt(encrypted_data).decode()


def hash_data(data): return
hashlib.sha256(data.encode()).hexdigest()


    key = generate_key()

    encrypted_text = encrypt_data("Sensitive
    Information", key)

    hashed_value = hash_data("Sensitive
    Information")


    decrypted_text = decrypt_data(encrypted_text,
    key)

    print(f"Encrypted: {encrypted_text}")

    print(f"Decrypted: {decrypted_text}")

    print(f"Hashed: {hashed_value}")
```

## 9. LIMITATIONS

- Homomorphic encryption requires high computational resources.
- Blockchain transaction speed may impact performance.
- Requires integration with existing cloud infrastructures.
- High initial setup cost for blockchain implementation.

## 10. RESULTS

The proposed model was tested using various security metrics. Results indicate improved data confidentiality and integrity compared to traditional encryption models.

| Security Feature | Traditional Cloud Storage | Owner Bot Model |
|---|---|---|
| Data Encryption | Standard AES/RSA | Homomorphic Encryption |
| Integrity Verification | Centralized Logs | Blockchain-based Logs |
| Unauthorized Access Prevention | Weak Authentication | Public-Private Key Pairs |

## 11. CONCLUSION

This research presents an advanced cloud security model integrating homomorphic encryption and blockchain technology. Future enhancements could include AI-driven threat detection and multi-cloud compatibility.

## 12. REFERENCES

1. Singh, A., & Singh, M. (2020). Cloud computing: Security and privacy issues. *International Journal of Computer Applications*.
2. Zhang, Y., & Liu, S. (2019). Cloud Storage Security Issues and Countermeasures. *Journal of Computer Security*.
3. Li, J., Wang, C., & Yu, S. (2017). Data Security in Cloud Computing. *International Journal of Cloud Computing*.
4. Zhao, X., & Li, W. (2018). Blockchain-based cloud data security. *International Journal of Cloud Computing*.
5. Patel, K., & Mishra, A. (2019). A survey on cloud storage security. *Procedia Computer Science*.