Robust Data Security Using Re-Encryptio And Access Revocation Mechanisms In Collaborative Platforms

Dr.A.Karunamurthy[1], V.Sujatha[2]
[1]Professor,Dept of computer Applications
[2]Dept of computer Applications
[1, 2] Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India

Abstract- *In modern collaborative environments, safeguarding shared data is critical. This project presents a secure file management system that employs onion encryption, dynamic re-encryption, and access revocation policies to ensure data confidentiality. The system automatically re-encrypts files and rotates encryption keys upon access or permission changes, rendering previously authorized data inaccessible to revoked users. RSA encryption is used to verify user identities before granting access, preventing unauthorized entry. A dedicated module manages periodic key rotation and integrity verification to detect and prevent data tampering. Crucially, all encryption and re-encryption operations occur without exposing plaintext, enhancing security during key updates or access changes. This solution offers robust protection against unauthorized access and ensures compliance with privacy standards, making it ideal for organizations seeking secure data collaboration. The development of this project leverages HTML and Bootstrap (CSS) for the front-end, ensuring a visually appealing and responsive design. On the backend, Java and MySQL have been employed to establish a robust foundation for data management and system functionality.*

Keywords- Onion encryption, Dynamic Re-Encryption, Access Revocation Policies, Data Confidentiality, RSA encryptions, Secure Data Collaboration, Periodic Key Rotation , Unauthorized access , Responsive Design

I. INTRODUCTION

In the era of digital transformation, collaborative platforms have become essential tools for organizations and individuals to share data, communicate, and work together efficiently. These platforms enable real-time access and collaboration from multiple locations, enhancing productivity and flexibility. However, this convenience comes at the cost of increased vulnerability to data breaches, unauthorized access, and misuse of sensitive information. The growing concern over data security in such environments highlights the need for stronger, more adaptive protection mechanisms to ensure the integrity and confidentiality of shared data.

Traditional security measures, such as static encryption and access control, often fall short in dynamic collaborative settings where users frequently change roles, join or leave teams, or require different access levels. To address these challenges, this project introduces a robust data security framework that integrates **re-encryption** and **access revocation** mechanisms. Re-encryption allows data to be re-secured with new encryption keys when access rights are updated or compromised, ensuring continued protection without exposing the original data. Access revocation mechanisms ensure that once a user's privileges are removed or modified, they can no longer retrieve or decrypt previously accessible data.

By combining these techniques, the proposed system enhances both data security and flexibility in collaborative platforms. It minimizes the risks associated with insider threats and unauthorized access, while also adapting to the dynamic nature of modern digital workspaces. This approach not only ensures the continuous protection of sensitive information but also fosters trust among users and organizations by providing stronger control over who can access what, when, and how.

II. LITERATURE SURVEY

Ensuring secure data sharing in collaborative platforms has been a central concern for researchers and developers alike. Traditional security models, including role-based access control (RBAC) and attribute-based encryption (ABE), have laid the groundwork for securing data access. However, these models often lack the ability to dynamically adapt to changing access permissions. Studies such as those by Sahai and Waters (2005) introduced **Attribute-Based Encryption (ABE)** as a flexible mechanism for enforcing access policies, but it does not inherently support efficient revocation or re-encryption, which are essential in dynamic collaborative environments.Recent advancements have explored the use of **Proxy Re-Encryption (PRE)**, as discussed by Ateniese et al. (2006), where a proxy can transform ciphertexts from one encryption key to another without accessing the plaintext. This approach allows for efficient delegation and revocation of access in distributed systems. Further research has built upon this by integrating PRE with decentralized key management and blockchain for auditability, as explored in works like Yu et al. (2010), which emphasize **fine-grained data sharing** and control in cloud settings.Additionally, literature has addressed **access revocation mechanisms**, such as key rotation and ciphertext update protocols, which ensure that previously granted access is effectively removed. Papers like “Efficient Revocation in Ciphertext-Policy Attribute Based Encryption” by Hur and Noh (2011) focus on reducing the overhead associated with revoking access in encrypted systems. Together, these contributions provide a strong foundation for the proposed project, which aims to merge re-encryption and revocation strategies to form a robust security framework suited for modern collaborative platforms, ensuring data remains secure despite changing user permissions.

III. PROBLEM STATEMENT

In today’s digital workspaces, collaborative platforms are widely used to facilitate communication, file sharing, and joint work among geographically dispersed users. These platforms handle a vast amount of sensitive and confidential information that must be protected from unauthorized access. However, conventional access control mechanisms and static encryption techniques often fall short in dynamic environments where users frequently change roles, join or leave groups, or require temporary access. Once data is shared or encrypted using a particular key, revoking a user’s access without compromising the security of the entire system becomes difficult. This creates a significant risk of data exposure, especially in cases of insider threats, accidental sharing, or external attacks.The lack of efficient and secure methods for re-encryption and access revocation creates challenges in maintaining data confidentiality and compliance with data protection regulations. Manual updates to encryption keys or access permissions are not only time-consuming but also prone to errors and inconsistencies. Therefore, there is a critical need for an intelligent and automated system that supports **re-encryption** to update encryption without revealing plaintext, and **access revocation** to instantly remove user access without affecting other legitimate users. The proposed project aims to fill this gap by designing a robust data security framework tailored for collaborative platforms, ensuring data remains protected even as access conditions evolve dynamically.

IV. PROPOSED SYSTEM

The proposed system is designed to offer strong data protection for users who share and manage files in a collaborative environment. It uses multiple security layers to make sure that sensitive information remains private and safe from unauthorized access. This system uses **onion encryption**, where data is locked using several layers of encryption. Even if one layer is broken, the others still protect the file. It also has **re-encryption** features, meaning whenever access rights are changed or a user is removed, the system will automatically re-encrypt the file using a new key. This prevents old users from viewing the file again. **RSA encryption** is used to check the identity of every user before giving access. This ensures that only trusted users can open or download the files. The system also includes **key rotation** and **file integrity checks** to make sure no one has changed the file without permission. If any problem is found, the system alerts the admin and updates the encryption immediately. Overall, this system is built to keep data secure, even when shared among many users.

Advantages of proposed system

- **Multi-Layered Encryption (Onion Encryption)** Files are encrypted using multiple layers, making it very hard for attackers to break through and access the actual data.
- **Automatic Re-Encryption After Access Change** When a user’s permission is changed or revoked, the file is re-encrypted with a new key, ensuring they can’t access the file again.
- **Secure User Verification with RSA** Every user must verify their identity using RSA encryption, which ensures that only trusted users can access the system.
- **Key Rotation for Better Security** Encryption keys are automatically changed at regular intervals, reducing the chance of key misuse.
- **File Integrity Checking** The system regularly checks if any file has been tampered with. If any changes are found, it takes action to secure the file again.

V. ARCHITECTURAL DESIGN

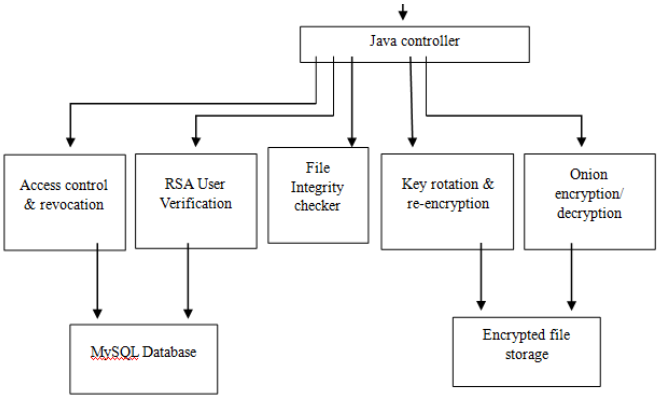

Fig 5.2: Architectural design

1. User Interface Layer

- **HTML + Bootstrap UI**:
  - The front-end interface that users interact with.
  - Built using HTML and Bootstrap for responsive design.
  - Allows users to upload/download files, login, and perform other actions.

2. Controller Layer

- **Java Controller**:
  - Acts as the central control logic between the frontend UI and backend services.
  - Receives user requests and routes them to the correct processing modules (e.g., access control, encryption).

Interfacing with Java Controller

1. **Access Control & Revocation**
   - Manages user permissions for accessing files.
   - Can revoke access when necessary.
   - Connected to the **MySQL Database** for user-role data.
2. **RSA User Verification**
   - Handles authentication using RSA public-key encryption.
   - Verifies the user's identity securely before granting access.
3. **File Integrity Checker**
   - Ensures the file hasn't been altered or corrupted.
   - Typically uses checksums or hash functions (e.g., SHA-256).
4. **Key Rotation & Re-encryption**
   - Automatically rotates encryption keys after a set period or event.
   - Re-encrypts files using new keys to maintain security over time.
   - Linked to **Encrypted File Storage**.
5. **Onion Encryption / Decryption**
   - Applies multi-layered encryption to files (like onion routing).
   - Enhances privacy and ensures that even if one layer is broken, data remains protected.

Data Storage

- **MySQL Database**:
  - Stores user credentials, access rights, and system logs.
- **Encrypted File Storage**:
  - Stores files in encrypted format.
  - Works with both the Onion Encryption and Key Rotation modules.

VI. TECHNOLOGIES USED

Java (Backend Logic & Security) Java is used as the primary backend language due to its strong security features, platform independence, and robust support for file handling and networking. In this project, Java is responsible for implementing core functionalities such as **AES encryption and decryption**, **QR code generation**, **user authentication**, and **interaction with the FTP-based cloud storage**. Its object-oriented nature also helps in building a modular and maintainable codebase.

MySQL (Database Management) MySQL is used as the **relational database system** to manage and store all critical information including **user credentials**, **file metadata**, **encryption keys**, and **access logs**. The database ensures data consistency, fast retrieval, and secure storage, supporting key operations such as user verification and tracking access activity.

HTML (Web Structure)HTML (HyperText Markup Language) forms the **foundation of the front-end interface**, defining the structure of web pages used for login, registration, file upload, and access control. It enables the integration of input forms, display elements, and links that connect users to system functionality.

CSS & Bootstrap (Styling & Responsiveness)CSS is used to style the web pages and improve the overall user experience. The project also uses **Bootstrap**, a popular CSS framework, to create a **responsive and mobile-friendly interface**. With its pre-built components and grid system, Bootstrap enhances visual appeal and ensures that the application functions well across different devices.

JavaScript (Interactivity & Frontend Logic) JavaScript adds **dynamic behavior and interactivity** to the front end. It is used for client-side form validation, QR code scanning and rendering, and asynchronous interactions. JavaScript ensures a smoother and more responsive user experience by minimizing page reloads and handling events in real time.

VII. PROPOSED TECHNIQUES

User Registration and Attribute Assignment:

- Each user is registered in the collaborative platform and assigned specific attributes (e.g., role, department).
- These attributes determine access rights using **Attribute-Based Encryption (ABE)**.

Initial Data Encryption:

- When a user uploads data to the platform, it is encrypted using ABE based on a defined access policy (e.g., only managers in the finance department).
- The data is stored in an encrypted format on the cloud or shared workspace.

Key Generation and Management:

- A **Key Management System (KMS)** securely generates, distributes, and stores encryption and re-encryption keys.
- Keys are periodically rotated to enhance security, and access is logged.

Secure Data Sharing:

- Users who meet the access policy (based on their attributes) are able to decrypt and access the data.
- If a new user needs access, a **Proxy Re-Encryption (PRE)** server securely re-encrypts the data for that user using their public key, without exposing the plaintext.

Dynamic Access Control:

- When a user's role or attribute changes (e.g., promotion, team switch), the system updates their access rights.
- If they no longer meet the access policy, they are removed from the decryption list.

Access Revocation:

- If a user is removed or their access must be revoked (e.g., they leave the organization), the system triggers a **re-encryption process**.
- The data is re-encrypted using a new key, and only valid users receive updated decryption keys.
- The revoked user cannot access either old or updated data.

Audit and Monitoring:

- All access, encryption, decryption, and re-encryption events are recorded in **audit logs**.
- Administrators can monitor for unusual activity and ensure compliance with data security policies.

Data Retrieval and Integrity Check:

- Authorized users can access and decrypt the data on demand.
- Integrity checks (e.g., using hashes) ensure that the data has not been tampered with.

VIII. CONCLUSIONS AND FUTURE ENHANCEMENTS

In conclusion, this project has successfully addressed the critical need for enhanced data security in collaborative platforms by implementing robust re-encryption and access revocation mechanisms. Through the integration of advanced cryptographic techniques, the system ensures that sensitive information remains protected, even in dynamic user environments where access permissions may change frequently. The project's objectives—to safeguard data integrity, maintain confidentiality, and provide flexible access control—have been met, demonstrating the feasibility and effectiveness of the proposed solutions.

The implementation of proxy re-encryption allows for secure data sharing without exposing the underlying plaintext, while the access revocation feature ensures that users no longer authorized can be promptly and effectively denied access. These features collectively contribute to a more secure and trustworthy collaborative environment, addressing common vulnerabilities associated with data sharing and user management. The system's performance under various scenarios has been evaluated, confirming its reliability and scalability for real-world applications.

Looking forward, future enhancements could focus on optimizing the system's performance, integrating machine learning algorithms for predictive access control, and extending compatibility with a broader range of collaborative tools. Additionally, conducting user experience studies can provide insights into usability improvements, ensuring that security measures do not impede user productivity. Overall, this project lays a solid foundation for advancing data security practices in collaborative settings, offering a viable solution to the challenges of secure data sharing and dynamic access management.

In this project, we have developed a secure file management system that helps protect sensitive data in collaborative platforms. The main focus was on providing strong security by using onion encryption, RSA-based user verification, and access revocation methods. The system ensures that only trusted users can access files, and once access is removed, the data is re-encrypted to prevent further viewing. We also added key rotation and file integrity checks to make sure the data is not tampered with and stays safe over time. These features work together to reduce the chances of unauthorized access and data leaks. Overall, this system is a useful solution for organisations and teams that need to share files securely and maintain control over who can access them. It improves data privacy and helps maintain trust in digital collaboration.

Future Enhancements

The Application can have the following enhancements. In this project, many future plans may be implemented to this process. However future thinking has expanded hugely in scope and substance. This system will expand to more future plans.

- **Biometric Verification**

  Adding fingerprint or face recognition to verify users can make the login process more secure and faster.

- **Real-Time Activity Monitoring**

  A feature to monitor user activities in real-time can help in quickly detecting any suspicious behaviour and preventing data misuse.

- **Cloud Integration**

  The system can be connected to trusted cloud services to allow secure file sharing and storage from anywhere.

- **Mobile App Support**

  Developing a mobile application will make it easier for users to access and manage files securely using their smartphones.

- **AI-Based Threat Detection**

  Artificial Intelligence can be used to detect unusual access patterns and automatically block or report suspicious users.

REFERENCES

[1] Stallings, William. *Cryptography and Network Security: Principles and Practice*. Pearson Education, 2017.

[2] Menezes, Alfred J., Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[3] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, vol. 21, no. 2, 1978, pp. 120–126.

[4] Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, 1996.

[5] Sandhu, Ravi, and Pierangela Samarati. "Access Control: Principles and Practice." *IEEE Communications Magazine*, vol. 32, no. 9, 1994, pp. 40-48