International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

TRUSTED WEB-BASED CLOUD STORAGE FOR SECURE DATA SHARING

Mangina Umamaheswararao[1], Ballanki Venu Gopal[2], Chinthapalli Venu Gopal [3], Manukonda Satish [4], Veedhi Uday[5], Valavala Surya Teja[6]

[1]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[2]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[3]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[4]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[5]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[6]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101

Abstract - *Public cloud storage, while offering cost reduction and accessibility, presents significant security risks due to potential data breaches and server-side vulnerabilities. Client-side encryption is a promising solution, but current implementations often suffer from limitations, including poor security due to weak password-based encryption, inflexible hybrid encryption schemes, and usability issues stemming from required software installations. This paper identifies the shortcomings of existing client-side encryption approaches, such as limited file-sharing capabilities and coarse-grained access control. To address these challenges, we propose a trusted web-based cloud storage system designed to enhance data protection, control unauthorized access, and facilitate secure file sharing while preserving data integrity and confidentiality. This system aims to provide a secure, efficient, and user-friendly solution, mitigating the security and usability drawbacks of current cloud storage encryption methods.*

***Key Words*: Public Cloud Storage, Cryptographic Techniques, Unauthorized Access Prevention, Encryption Algorithms, Secure File Sharing**

1. INTRODUCTION

Public cloud storage services have gained widespread adoption due to cost efficiency and ease of use, prompting individuals and organizations to store and share data online. However, this reliance on cloud providers raises significant security concerns, as sensitive data is often stored unencrypted, making it vulnerable to breaches and unauthorized access. While cloud providers implement server-side encryption, transport-layer security (TLS), and authentication mechanisms, these measures do not fully eliminate the risks associated with insider threats, misconfigurations, or cyberattacks.

To mitigate these risks, **client-side encryption** has emerged as a robust solution, ensuring that data is encrypted before being uploaded and only decrypted by authorized users. Since the cloud provider stores only encrypted data, exposure from server-side vulnerabilities is minimized. However, major cloud storage providers like Google Drive and Dropbox do not support built-in client-side encryption, relying instead on server-side measures. Apple iCloud employs end-to-end encryption only for select data types, leaving most stored information vulnerable to potential breaches.

Several encryption techniques have been employed to enhance cloud security. **Password-based encryption**, often utilizing symmetric encryption algorithms like AES, is a common approach. However, it suffers from vulnerabilities due to low-entropy passwords, making brute-force attacks feasible. Additionally, such methods typically support only single-user encryption, limiting secure file-sharing capabilities.

Hybrid encryption, integrating a **Key Encapsulation Mechanism (KEM)** and a **Data Encapsulation Mechanism (DEM)**, is another widely used approach. Public cloud providers such as Amazon, Tresorit, and Mega implement the RSA-AES model, where data is encrypted using AES, and encryption keys are secured with RSA public keys. However, this approach has drawbacks, including inefficiencies in managing recipient public keys, increased cipher text size, and higher computational costs when sharing data with multiple users.

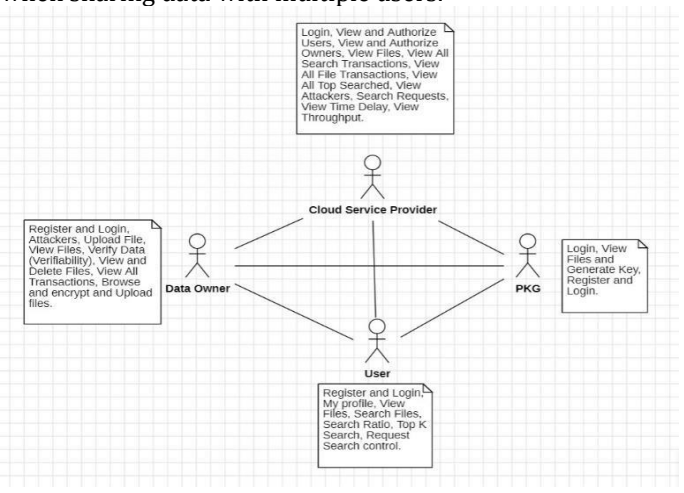

Figure 1.1: System Architecture

1.1 Benefits:

Enhanced Data Security

- Encrypting data **before uploading** ensures that only authorized users can access the original content.
- Eliminates reliance on **cloud provider trust**, reducing risks of data breaches from insider threats or external attacks.
- Protects against unauthorized access, even if the cloud provider's security is compromised.

Protection Against Data Leakage

- Since the cloud stores only **encrypted data**, exposure from misconfigurations, hacking attempts, or legal surveillance is significantly reduced.
- Unlike traditional cloud storage, where providers have access to user data, **client-side encryption prevents unauthorized decryption**.

Improved Access Control and File Sharing

- Unlike traditional **password-based encryption**, which supports only single-user encryption, your project enables **secure multi-user file sharing**.
- Ensures **fine-grained access control**, allowing different users to have different permissions without exposing encryption keys.

Efficient and Scalable Encryption Mechanism

- Overcomes inefficiencies of existing **RSA-AES hybrid encryption**, where increasing recipients results in larger cipher text and higher computation costs.
- Provides a **lightweight and scalable** encryption model that optimizes storage and bandwidth usage.

Better Usability and Platform Compatibility

- Eliminates the need for **third-party software or plugins**, improving accessibility across multiple devices and platforms.
- Users do not need to repeatedly **reinstall encryption software**, making data sharing and storage **more seamless and user-friendly**.

Cost Reduction and Performance Optimization

- Unlike enterprise-grade encryption solutions that demand **high processing power and expensive infrastructure**, your project offers a cost-effective alternative.
- Reduces the burden on cloud providers by shifting encryption processes to **client-side devices**, minimizing **storage and computational overhead**.

Increased Trust and Compliance

- Addresses compliance concerns for **GDPR, HIPAA, and other data privacy regulations**, ensuring users maintain **full control over their sensitive information**.
- Boosts user confidence in cloud storage by eliminating reliance on **third-party trust**.

2. LITERATURE SURVEY

The literature review stands out as a crucial phase in the software development process. It entails an exploration of prior research conducted by various authors in the relevant field. We will analyze and build upon key articles to enhance our work.

2.1 A Secure Framework for Data Storage and Sharing in Cloud Computing (2019) This paper proposes a secure framework for storing and sharing data in cloud computing while ensuring confidentiality and integrity. It employs **client-side encryption** to protect data before uploading and decrypts it after downloading, preventing unauthorized access. The framework incorporates **access control mechanisms**, allowing only authorized users to retrieve and share data securely. Additionally, it ensures **data integrity** through cryptographic techniques, reducing the risk of data tampering or unauthorized modifications. The proposed solution enhances **security, efficiency, and usability**, making cloud storage more reliable.**Mohammad Reza** In this paper, a speech emotion recognition system based on a 3D CNN is suggested to analyse and classify the emotions. He concluded, the three-dimensional reconstructed phase spaces of the speech signals were calculated in order to recognise the emotion in speech.

2.2 Group Key Management Protocol for File Sharing on Cloud Storage (2020) presents a **group key management protocol** to facilitate secure file sharing on cloud storage. It ensures that only **authorized group members** can access shared files while preventing unauthorized access. The protocol efficiently manages **dynamic group membership** by securely distributing and updating encryption keys. It reduces **computational overhead and communication costs**, making file sharing **more efficient and scalable**. The proposed approach enhances **security, confidentiality, and access control** in cloud-based file-sharing systems.

2.3 Secure Data Sharing Mechanism for Cloud-Edge Computing (2023) introduces a **secure data-sharing mechanism** designed for **cloud-edge computing** environments. It integrates **encryption and access control techniques** to protect sensitive data while enabling **efficient sharing** between cloud and edge nodes. The mechanism ensures **low-latency data access** and minimizes computational overhead, making it suitable for **real-time applications**. Additionally, it addresses **key management challenges**, maintaining security while supporting **dynamic user access**. The proposed solution enhances **data confidentiality, integrity, and performance** in cloud-edge computing systems.

2.4 A Secure Cloud Storage Framework with Enhanced Integrity and Auditability Using Consortium Blockchain System (2024) presents a **secure cloud storage framework** that improves **data integrity and auditability** using a **consortium blockchain system**. It ensures that stored data remains **tamper-proof** by recording transactions on a **decentralized ledger**. The framework enables **verifiable audits**, allowing users to check data integrity without relying on a third party. It also integrates **encryption and access control mechanisms** to prevent unauthorized access. The proposed solution enhances **security, transparency, and trust** in cloud storage environments.

2.5 A Reliable Approach for Data Security Framework in Cloud Computing Network (2024) proposes a reliable

data security framework for cloud computing networks to protect sensitive information from unauthorized access and cyber threats. It integrates encryption, authentication, and access control mechanisms to enhance data confidentiality and integrity. The framework ensures secure data transmission and storage while minimizing vulnerabilities to attacks. It optimizes performance by reducing computational overhead and improving scalability. The proposed approach strengthens cloud security, reliability, and user trust.

2.6 Web Cloud Internet Linked Cloud Repository for Secure Exchange of Data (2024) introduces a Web Cloud Internet Linked Cloud Repository (WCILCR) for the secure exchange of data. It ensures data confidentiality and integrity using encryption and access control mechanisms. The system enables seamless and efficient data sharing across multiple cloud platforms while preventing unauthorized access. Additionally, it incorporates auditability features to track and verify data exchanges. The proposed solution enhances security, transparency, and interoperability in cloud-based data-sharing environments.

3. EXISTING SYSTEM

Existing cloud storage systems are often insecure due to a lack of strong client-side encryption. They rely on server-side encryption, which puts data at risk, or use complex cryptographic methods that are inefficient or impractical for web applications. Furthermore, solutions requiring plugins limit usability. Attribute-Based Encryption (ABE) offers better access control but is computationally heavy and lacks immediate user revocation, creating security vulnerabilities

3.1 Relatively Poor Security: The absence of end-to-end encryption leaves cloud data susceptible to breaches, as information is vulnerable both during transmission and once stored on the server. Furthermore, the lack of a reliable revocation mechanism for shared data exacerbates security risks, as access rights cannot be effectively managed or revoked, potentially leading to unauthorized data exposure.

3.2 Coarse-grained Access Control: Existing cloud storage suffers from rigid access controls, restricting precise sharing policies and limiting user control over data access, which results in inefficient file-sharing practices.

3.3 Poor usability: Many existing solutions require additional browser extensions or plugins, creating compatibility issues across different devices and operating systems. Complex encryption and decryption processes add overhead, making it difficult for non-technical users to securely store and manage files.

3.4 Lack of Immediate User Revocation: When a user's access rights change or need to be revoked, **there is no immediate enforcement,** creating security risks. Delayed revocation increases the chances of unauthorized access to sensitive data.

4. PROPOSED WORK

Trusted Web-Based Cloud Storage offers a user-friendly, secure cloud solution with client-side encryption and fine-grained access control via optimized CP-ABE. It improves upon traditional ABE with features like immediate revocation and offline encryption, providing strong security and performance. Built on ownCloud, it delivers fast, scalable, and secure storage without requiring plugins. **High-Quality and Diverse Datasets:** A critical foundation for any speech emotion analysis system is a comprehensive, high-quality dataset that covers a wide range of emotions and demographic groups. Collecting diverse data can help reduce biases and improve the system's generalizability.

- **End-to-End Client-Side Encryption:** Data is encrypted on the user's device before it's uploaded and decrypted only after download, ensuring that the cloud provider never sees unencrypted data, thus maximizing data confidentiality.
- **Fine-Grained Access Control:** Users can precisely define who can access specific data, allowing for highly controlled sharing and preventing unauthorized access to sensitive information.
- **Immediate User Revocation:** Access rights can be instantly revoked, preventing unauthorized access when user permissions change, thus reducing security vulnerabilities.
- **Offline Encryption Support:** Encryption and decryption can be performed even without an internet connection, improving usability and ensuring data protection in various scenarios.

- **Outsourced Decryption for Performance Optimization:** Decryption tasks can be offloaded to a third party, reducing the computational load on user devices and improving performance, especially on resource-constrained devices.
- **Cross-Platform and Plugin-Free:** The system works across different operating systems and devices without requiring browser extensions or plugins, enhancing accessibility and usability for all users.
- **Enhanced Security Model:** A robust security model with provable security against web and cryptographic threats ensures a high level of protection for stored data.

METHODOLOGY:

This project employs a hybrid encryption strategy by integrating AES (Advanced Encryption Standard) for data confidentiality and RSA (Rivest-Shamir-Adleman) for secure key transmission. The process encompasses key generation, encryption, and decryption to ensure secure storage and sharing of data in a cloud environment

1. Cryptographic Key Generation:

The encryption framework begins by generating necessary cryptographic keys:

- **AES Key**: A symmetric key is generated using a fixed string or a secure random generator. This key, typically 128 or 256 bits in length, is responsible for encrypting the actual data quickly and securely.
- **RSA Key Pair:** An RSA key pair (public and private keys) is generated using Java's KeyPairGenerator class. The public key is later used to encrypt the AES key, while the private key is reserved for decrypting it on the receiver's side.

2. Encryption Workflow:

The encryption process is divided into two phases:

Phase 1: Encrypting Data using AES

1. The plain text is encrypted using the AES algorithm in encryption mode.
2. The encrypted output is encoded using Base64 to ensure it can be safely transmitted or stored in textual form.

Phase 2: Securing the AES Key using RSA

1. The AES key, in its byte array form, is encrypted using the recipient's RSA public key.
2. This ensures that only someone with the corresponding RSA private key can retrieve and use the AES key.
3. The encrypted AES key is also encoded in Base64 for seamless transmission.

3. Decryption Process:

Decryption is carried out in the reverse order:

Step 1: Recovering AES Key via RSA Decryption

1. The AES key is retrieved by decrypting the Base64-decoded RSA-encrypted key using the private RSA key.
2. This operation ensures that only the intended recipient can gain access to the encryption key.

Step 2: Decrypting Encrypted Data using AES

1. Using the decrypted AES key, the encrypted content is decrypted back to its original plain text form.
2. The result is the original message that was initially encrypted.

4. Security Measures and Performance Considerations:

- **Hybrid Encryption Advantage:** The system combines the speed and efficiency of AES with the robust key security of RSA. AES handles large data volumes, while RSA secures the exchange of AES keys.
- **Base64 Encoding:** Encoding binary encrypted data into Base64 helps ensure compatibility during transmission or storage.
- **Resource Optimization:** Since RSA is only used for encrypting small amounts of data (i.e., keys), the computational load is minimized. AES handles the bulk data processing efficiently.

6.User Interface:

Figure 6.1: User Interface

7.CONCLUSION:

This hybrid cryptographic system leverages the strengths of both AES and RSA. AES secures the actual content, while RSA safeguards the encryption key, offering a secure and efficient method for data storage and sharing. This approach provides a balanced solution, maintaining both performance and data confidentiality in cloud-based systems.

8. REFERENCES:

[1] Web Cryptography API, The Web Cryptography WG of the W3C, Tech. Rep., January 2017. Provides a standard for performing cryptographic operations within web applications.

https://www.w3.org/TR/WebCryptoAPI/

[2] E. Stark, M. Hamburg, and D. Boneh, "Symmetric cryptography in JavaScript," in *Computer Security Applications Conference (ACSAC'09)*, IEEE, 2009, pp. 373–381

https://ieeexplore.ieee.org/document/5375889

[3] OpenPGP.js, "OpenPGP Implementation for JavaScript," [Online]. A JavaScript library for client-side encryption, useful for implementing web-based encryption solutions.

https://github.com/openpgpjs/openpgpjs

[4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International Workshop on Public Key Cryptography*, Springer, 2011, pp. 53–70. Introduces Ciphertext-Policy Attribute-Based Encryption (CP-ABE), essential for access control in cloud storage.

https://doi.org/10.1007/978-3-642-19379-8_4

[5] W. Zhu, J. Yu, T. Wang, P. Zhang, and W. Xie, "Efficient attribute-based encryption from R-LWE," *Chinese Journal of Electronics*, vol. 23, no. 4, pp. 778–782, 2014. Discusses efficient attribute-based encryption for secure data sharing in cloud environments.

https://doi.org/10.1049/cje.2014.08.007

[6] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *International Workshop on Public Key Cryptography*, Springer, 2014, pp. 293–310. Explores advanced encryption techniques for optimizing security and performance in cloud-based storage.

https://doi.org/10.1007/978-3-642-36362-7_18

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute-based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 261–270. Provides a framework for managing access control and attribute revocation in encrypted cloud storage systems.

https://doi.org/10.1145/1755688.1755723

[8] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, et al., "Bringing the web up to speed with WebAssembly," in *ACM SIGPLAN Notices*, vol. 52, no. 6, ACM, 2017, pp. 185–200. Discusses WebAssembly's role in improving encryption speed in web-based applications.

https://doi.org/10.1145/3140587.3062363

[9] R. Zhang, H. Ma, and Y. Lu, "Fine-grained access control system based on fully outsourced attribute-based encryption," *Journal of Systems and Software*, vol. 125, pp. 344–353, 2017. Explores fine-grained access control mechanisms relevant for blockchain-integrated cloud storage.

https://doi.org/10.1016/j.jss.2016.11.027

[10] OwnCloud, "OwnCloud - The leading open-source cloud collaboration platform, A widely used open-source cloud storage platform that supports client-side encryption.

https://owncloud.org/

[11] E. Bocchi, I. Drago, and M. Mellia, "Personal cloud storage: Usage, performance and impact of terminals," in 4th IEEE International Conference on Cloud Networking,

CloudNet 2015, Niagara Falls, ON, Canada, October 5-7, 2015. IEEE, 2015, pp. 106–111.

https://doi.org/10.1109/CloudNet.2015.7335291

[12] M. Green, S. Hohenberger, B. Waters et al., "Outsourcing the decryption of ABE ciphertexts," in USENIX Security Symposium, vol. 2011, no. 3, 2011.

https://www.sciencedirect.com/science/article/pii/S0167404819300525