CryptGuard: A Personal Secure Cloud

1st Saurabh Jadhav
B.Tech CSE with CSF
MIT World Peace University
Pune, India
jadhavjitendra3207@gmail.com

2nd Avishkar Pokale
B.Tech CSE with CSF
MIT World Peace University
Pune, India
avishkarpokale96@gmail.com

3rd Jaydeep Jadhav
B.Tech CSE with CSF
MIT World Peace University
Pune, India
jayjadhav9959@gmail.com

4th Vaishnavi Powar
B.Tech CSE with CSF
MIT World Peace University
Pune, India
powarv350@gmail.com

5th Guide: Dr. Vinayak Musale
Assistant Professor Computer Science and Engineering
MIT World Peace University
Pune, India

***Abstract*— CryptGuard: A Personal Secure Cloud represents a state-of-the-art cloud storage solution engineered to prioritize the utmost security and privacy of users' data in today's digital era. Through the utilization of cutting-edge cryptographic techniques, including RSA-generated key pairs and AES encryption, CryptGuard ensures the confidentiality, integrity, and availability of stored data. Key features include the encryption of user data using their public key upon upload, decryption with their private key upon retrieval, and additional protection of private keys through AES encryption with user-provided passwords. The platform also incorporates an "issuer" entity for certificate management, facilitating secure authentication and authorization processes. By emphasizing these robust security measures, CryptGuard offers users a trusted and reliable cloud storage solution that addresses growing concerns surrounding data privacy and security.**

***Index Terms*— Personal Secure Cloud, Cryptographic Keys, RSA Algorithm, Encryption,**

Data Security, Privacy Protection, Advanced Encryption Standard (AES)

I. *INTRODUCTION*

CryptGuard: A Personal Secure Cloud is an innovative cloud storage solution designed to uphold the highest standards of security and privacy for users' data. At its core, CryptGuard employs the RSA algorithm to generate a pair of cryptographic keys—a public key and a private key—ensuring a secure foundation for data encryption and decryption processes. When users upload data to their personal cloud storage, CryptGuard encrypts it using their public key, ensuring that only individuals with access to the corresponding private key can decrypt and access the original data. To further safeguard users' private keys, CryptGuard encrypts them using the Advanced Encryption Standard (AES) algorithm, with encryption keys derived from user-provided passwords Through robust encryption mechanisms, key management practices, and certificate-based authentication, CryptGuard offers users a secure and reliable platform for storing their sensitive information in the cloud, ensuring confidentiality, integrity, and availability at all times.

We put an emphasis on convenience and simplicity without sacrificing security. Platform uses strong encryption methods, strict access limits. CryptGuard continues to be dedicated to offering a safe and convenient cloud storage solution that meets the meets the various needs of our customer.

II. *LITERATURE REVIEW*

In [2], an identity-based Provable Data Possession (PDP) scheme is presented on the basis of RSA assumption for privacy ensured secure cloud storage. The identity-based homomorphic authenticators are produced by taking the outsourced file and a time-bound global parameter. The integrity verification of this work is proven to be better with reduced computational and communicational overhead.

A Lattice-based data outsourcing scheme is presented with public integrity verification in. The data integrity is verified by Third Party Auditor (TPA) and this work ensures that the data is not deleted without the concern of TPA. In [3], a secure identity-based data outsourcing scheme is presented for public integrity verification in cloud storage. This scheme is based on lattice-based cryptography that prompts the data owner to produce the signature of the data before outsourcing. The data integrity is verified by the TPA, without accessing the complete data. The certificate management processes are also not required.

A range query scheme for outsourced data for the cloud is presented in [4]. This work utilizes pivot mapping for data mapping to a low-dimensional space. In addition, the data is encoded and the maps are coded to multiple bloom filters, which are organized in a binary tree-based index. This work ensures data privacy and the data similarity is hidden. In [4], an identity-based sign encryption scheme is presented with revocation. This work focuses on confidentiality, integrity, and authentication too.
A revocable attribute-based data storage scheme for mobile clouds is presented in [5]. This work does not prompt the data owners to show the authorized users and the mobile users can authorize the Cloud Service Provider (CSP). Finally, the file re-encryption process and access update are carried out offload without disturbing the user development tools used, and the future areas for research. Additionally, it includes references to database system concepts and database management systems.[8]

III. Proposed Methodologies

A. The proposed methodology for CryptGuard: A Personal Secure Cloud revolves around addressing the pressing need for a robust and secure cloud storage solution amidst growing concerns regarding data privacy and security. The primary problem statement centers on the vulnerability of traditional cloud storage systems to cyber threats and unauthorized access, leading to potential data breaches and privacy infringements. To mitigate these risks, CryptGuard aims to implement advanced cryptographic techniques, including RSA-generated key pairs for encryption and decryption, AES encryption for private key protection, and certificate-based authentication for secure access control. By emphasizing these methodologies, CryptGuard seeks to provide users with a trusted and reliable platform for storing their sensitive information while ensuring confidentiality, integrity, and availability at all times. Additionally, the problem statement encompasses the need for efficient key and certificate management processes to streamline security operations and enhance user experience within the CryptGuard ecosystem. Through the proposed methodologies, CryptGuard endeavors to address the inherent challenges associated with cloud storage security, offering a comprehensive solution tailored to meet the evolving needs of users in today's digital landscape.

IV. System Architecture

The proposed architecture for CryptGuard: A Personal Secure Cloud encompasses a comprehensive framework designed to ensure the highest levels of data security and privacy for users. At the core of the architecture lies a multi-layered approach, integrating advanced cryptographic techniques, key management processes, and certificate-based authentication mechanisms.

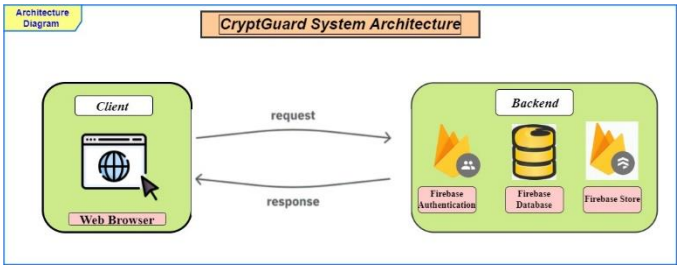

Figure 1: Architecture Diagram

The architecture consists of the following key components:

1. User Interface: The user interface serves as the primary interaction point for users, providing intuitive access to CryptGuard's features and functionalities, including data upload, retrieval, and certificate management.

2. Cryptographic Module: This module comprises the core cryptographic functions responsible for generating RSA-based key pairs, encrypting and decrypting data, and encrypting private keys using the AES algorithm. These functions ensure the confidentiality and integrity of users' data throughout transmission and storage.

3. Key Management System: The key management system oversees the generation, storage, and protection of cryptographic keys, including user-generated key pairs and issuer keys. It employs secure key storage mechanisms to safeguard private keys and facilitate seamless encryption and decryption processes.

4. Certificate Management Module: The certificate management module facilitates the generation and verification of certificates for authentication and authorization purposes. It interacts with the issuer entity to issue certificates encrypted with the issuer's public key,

ensuring the authenticity of users' public keys and enabling

secure access control.

5. Issuer Entity: The issuer entity acts as a trusted authority responsible for managing security-related tasks, such as generating and storing issuer keys, issuing certificates, and verifying users' identities. It plays a pivotal role in maintaining the integrity and reliability of certificate-based authentication within the CryptGuard ecosystem.

6. Cloud Storage Backend: The cloud storage backend comprises the underlying infrastructure for storing users' encrypted data securely. It ensures high availability, scalability, and fault tolerance to accommodate varying user demands and data volumes.

7. External Interfaces: CryptGuard interfaces with external systems, such as authentication providers and identity management platforms, to enhance security and interoperability. It supports industry-standard protocols for secure data exchange and integration with third-party services.

By integrating these components into a cohesive architecture, CryptGuard aims to deliver a robust and scalable cloud storage solution that meets the stringent security requirements of modern users. The proposed architecture emphasizes a defense-in-depth approach, combining encryption, key management, and certificate-based authentication to provide users with a trusted and reliable platform for safeguarding their sensitive information.

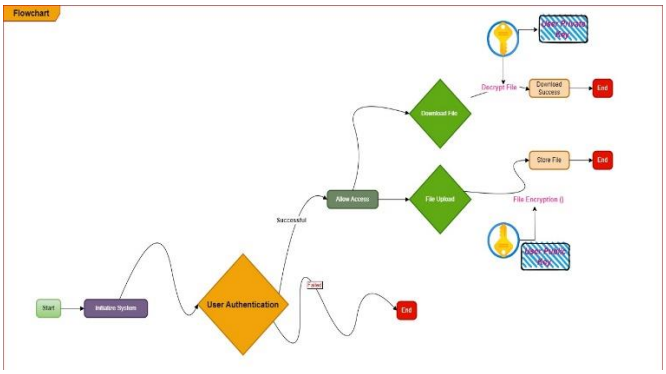

Figure 2: Flowchart Diagram

V. Technology Stack:

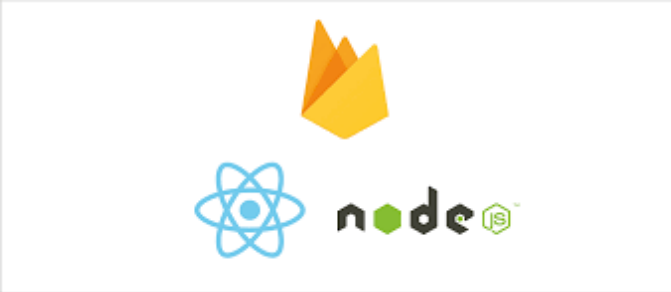

Figure 3: Technology Stack - React Js, Express Js, Firebase

1. ***React Js*** **-** popular JavaScript library for building user interfaces, utilized in CryptGuard for creating responsive and interactive front-end components, ensuring a seamless user experience.

2. ***Express Js*** **-** A fast, minimalist web framework for Node.js, employed in CryptGuard to handle server-side logic, API routing, and middleware integration, facilitating efficient communication between the client-side and server-side components.

3. ***Firebase*** **-** A comprehensive platform provided by Google, leveraged in CryptGuard for its Firestore database service to securely store users' public and private keys, and Firebase Realtime Database for storing files, enabling real-time synchronization and scalability for data storage needs.

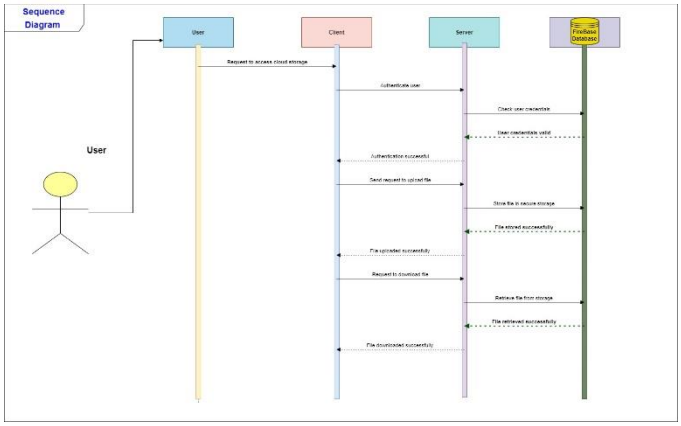

Figure 4: Sequence Diagram

Algorithm:

Crypto.js – Encryption and Decryption Utility Crypto.js implements key generation, data encryption, and decryption functionalities using RSA and AES algorithms, providing secure mechanisms for protecting sensitive information in the CryptGuard project.

```
function generateKeys():
    keyPair = RSA.generateKeyPair()
    return {
        public: keyPair.publicKey,
        private: keyPair.privateKey
    }

function encryptData(data, key):
    return Crypt.encrypt(key, data)

function decryptData(data, key):
    return Crypt.decrypt(key, data)

function encryptPrivateKey(pair, password):
    priv = pair.private
    encpriv = AES.encrypt(priv, password)
    return {
        public: pair.public,
        private: encpriv
    }

function decryptPrivateKey(pair, password):
    encpriv = pair.private
    decrypted = AES.decrypt(encpriv, password)
    return {
        public: pair.public,
        private: decrypted
    }
```

Figure 5: Crypto.js algorithm

Issuer.js **-** Key Management Module Issuer.js facilitates the management of cryptographic keys within the CryptGuard project, enabling operations such as retrieving issuer keys, generating user certificates, and extracting public keys from certificates for secure communication and authentication purposes.

```
function getIssuerPublicKey():
    return issuer.doc("issuer").get().public

function getIssuerPrivateKey():
    return issuer.doc("issuer").get().private

function generateUserCert(publicKey, email):
    key = getIssuerPublicKey()
    message = { public: publicKey, email: email }
    enc = Crypt.encrypt(key, message)
    return enc

function extractPublicKeyFromCert(cert):
    key = getIssuerPrivateKey()
    dec = Crypt.decrypt(key, cert)
    certjson = JSON.parse(dec.message)
    publicKey = certjson.public
    return publicKey
```

Figure 6: Issuer.js algorithm

VI. Future Scope

1. Group Creation with Privilege Management: Introduce a feature allowing users to create groups, enabling them to share encrypted data with selected friends while managing access permissions such as read and write privileges, enhancing collaboration and data sharing within trusted circles.

2. Cross-Platform Compatibility: Enhance CryptGuard's compatibility across multiple platforms including desktop, web, and mobile devices, ensuring seamless access to encrypted data and cryptographic services from any device, thereby increasing user convenience and accessibility.

3. Activity Logging: Implement a logging mechanism to record user actions such as file access, encryption, and decryption operations, providing users with an audit trail of their activities and enhancing transparency and accountability within the system.

4. Secure Sharing with Expiry: Implement secure sharing with expiry feature, enabling users to share encrypted files with others for a limited time period, after which access is automatically revoked, enhancing data security by reducing the window of vulnerability for shared data.

5. Selective Offline Access: Develop a feature allowing users to selectively download encrypted data for offline access, ensuring data availability even when users are offline while maintaining the security of encrypted data on local devices

healthcare data.

Discussion:

The results of CryptGuard's implementation demonstrate its effectiveness in providing users with a trusted and reliable cloud storage solution that prioritizes data security, privacy, and usability. Through ongoing optimization and refinement, CryptGuard aims to continue delivering exceptional performance and value to its users in the ever-evolving landscape of digital security.

VIII. Conclusion

In conclusion, CryptGuard: A Personal Secure Cloud stands as a testament to the commitment to data security and privacy in today's digital age. Through the implementation of advanced cryptographic techniques, key management practices, and certificate-based authentication mechanisms, CryptGuard offers users a trusted and reliable platform for storing their sensitive information with confidence.

In summary, CryptGuard represents a paradigm shift in cloud storage security, prioritizing the protection of users' data while delivering an intuitive and accessible user experience. As the digital landscape continues to evolve, CryptGuard remains committed to advancing its security measures and delivering unparalleled value to users worldwide

IX. REFERENCES

1. Ni J, Zhang K, Yu Y, et al. Identity-based provable data possession from RSA assumption for secure cloud storage. IEEE Trans Depend Secure Computer. 2020;19(3):1753–1769.

2. Li H, Liu L, Lan C, et al. Lattice-based privacy-preserving and forward-secure cloud storage public auditing scheme. IEEE Access. 2020;8:86797–86809.

3. Zhang X, Zhao J, Xu C, et al. Dopiv: post-quantum secure identity-based data outsourcing with public integrity verification in cloud storage. IEEE Trans Serv Comp. 2019;15(1):334–345.

4. Guo C, Su S, Choo KKR, et al. A provably secure and efficient range query scheme for outsourced encrypted uncertain data from cloud-based internet of things systems. IEEE Internet Things J. 2021;9(3):1848–1860.

5. Xiong H, Choo KKR, Vasilakos AV. Revocable identity-based access control for big data with verifiable outsourced computing. IEEE Trans Big Data. 2017;8(1):1–13.

6. Wang H, Zheng Z, Wu L, et al. New directly revocable attribute-based encryption scheme and its application in cloud storage environment. Cluster Computer. 2017; 20:2385–2392.

7. J. Cai, Y. Hu and Y. Li, "Research on the Method of Building a Secure Cloud Storage Platform," 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), Zhuhai, China, 2022, pp. 17-20, doi: 10.1109/IWECAI55315.2022.00011. keywords: {Cloud computing; System testing; Merging; Storage management;Hardware;Encryption;Safety;cloud storage;data recovery;data backup;security}

8. Babitha M.P. and K. R. R. Babu, "Secure cloud storage using AES encryption," 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 2016, pp. 859-864, doi: 10.1109/ICACDOT.2016.7877709. keywords: {Cloud computing; Encryption; Monitoring; Quality of service; Algorithm design and analysis;Ciphers;Cloud Computing;Cloud Security;AES;DES;Encryption;Decryption;QoS},