ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Secure Outsourcing and Sharing of Cloud Data Using a User-Side Encrypted File System

U.Sai Laxmi
21N81A6269
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
usailakshmi05@gmail.com

N.Sai kiran
21N81A6293
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
ngvarma29@gmail.com

S.Vaishnavi
21N81A6268
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
shamolavaishnavi@gmail.com

T. Lehar Balaji
21N81A6272
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
balajithogata745@gmail.com

Mr.Surya Narayana Reddy
Assissant professor
Computer Science and Engineering (Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

ABSTRACT:

Cloud computing is an emerging paradigm that aims to provide computing resources, massive data storage capacity and flexible data sharing services. The explosive growth of data produced persuade business and users, driven by the cloud-top features, to outsource their data to the cloud storage systems. However, the confidentiality and integrity of outsourced sensitive data in remote cloud servers are becoming a major concern. Data must be encrypted prior to storing it in the potentially untrustworthy cloud. Existing traditional encryption systems impose a heavy burden of managing files and encryption operations on data owners. They suffer from serious security, efficiency and usability issues, and some schemes are inappropriate for protecting cloud data. In this application, we introduce OutFS, a user-side encrypted file system, focused on providing a transparent encryption for stored and shared outsourced data. In order to ensure robust data sharing security, the identity-based encryption scheme (IBE) is integrated with OutFS. OutFS is designed to preserve the integrity of outsourced file data and file system data structure.

INTRODUCTION:

Cloud computing is emerging as the most suitable paradigm for individuals and organizations to access inexpensive, scalable, ubiquitous, and on-demand computing resources, applications, and data storage services. Cloud storage systems, such as Dropbox, Google Drive, Apple's iCloud, Microsoft OneDrive, etc., enable users to remotely store a large volume of data that can be accessed and shared among users, regardless of time and location constraints.

With the growing popularity of cloud computing, the number of enterprises and individuals shifting toward the use of cloud has increased rapidly. As a result, a vast amount of important personal information and critical organization data, such as personal health records, government documents, and company finance data, etc., are transmitted across the Internet and stored in cloud servers. However, outsourcing sensitive data suffers from critical security threats, privacy, and access control problems. These are common concerns of organizations and individuals using the associate editor coordinating the review of this manuscript and approving it for publication was Parul Garg. cloud services. When data owners migrate their sensitive data to the cloud, they lose an element of control over their data. Cloud users have no guarantee about the way these sensitive data will be treated and protected by cloud providers. Although the cloud provides users with the convenience of data access across multiple devices, by using cloud services, user data are vulnerable to a verity of malicious attacks and threats. Security incidents occur frequently. Even worse, cloud service provider may leak user data to unauthorized entities for illegal profit.

One feasible solution to overcome these problems is to use cryptography. All sensitive data have to be encrypted by data owners prior to storing them into the potentially untrustworthy cloud. The strength of the encryption scheme is largely dependent on the strength of the key management technique used. The security of the encryption scheme lies on the secrecy of the keys that are known only to the users authorized to read their respective data, and not only on the secrecy of the encryption algorithm used.

Given the amount of data being stored and shared in cloud and the increasing number of data users, designing a cryptographic scheme for cloud storage that meets the requirements of security, efficiency, ease of use, and flexibility is a challenging task [6]. Traditional encryption applications, generally, suffer from limited usability due to the manual solution provided by applications. Data owners must

encrypt their data manually prior to uploading to the cloud. Moreover, users have to manually generate, manage, and store the encryption keys.

However, the involvement of data owners in performing multiple encryption and decryption operations is cumbersome and time consuming. Also, it is difficult for users to manage more than a few keys, and if the keys are leaked or otherwise compromised, security will be threatened. Encryption applications are designed to be bandwidth-hungry and latency- sensitive, in which the increased number of outsourced files requiring encryption would significantly affect the system performance and data access response time.

Recent works cope with the limitations of the encryption applications by adopting a transparent encryption approach. This type of encryption mechanism is implemented most effectively with the help of operating system file systems. The common approach is composed of a client application that interacts with the local cryptographic file system, and the encrypted data are synchronized or backup to connected back-end cloud storage servers. The cryptographic file system can be built as a layer inside the kernel space of the operating system. It can also be implemented as a file system in user space using FUSE technology to build a cryptographic file system on top of local or remote storage systems. Other schemes use a cloud- backed file system, such as S3FS, BlueSky, S3QL, which are implemented on remote file systems to provide encryption services to stored data in one or multiple clouds.

Existing cryptographic file system proposals have focused on transparent encryption of data-at-rest. However, the direct employment of these encryption schemes over remotely stored data cannot achieve the data security required in multi-user environments. Moreover, most of these approaches impose many restrictions on providing reliable and secure cloud data sharing, as they severely limit users to selectively sharing their data at a fine-grain level. In addition, these encryption file systems are based on the use of passwords to identify legitimate users and to retrieve the encryption keys. Other problems related to the cloud-backed file systems include performance bottlenecks, processing latency, bandwidth cost, and being a single point of failure.

When a data owner wants to grant decryption access to authenticated users, the data owner either exchanges the encryption keys manually, out-of-band or, encrypts individual shared data with the public keys of the respective users using asymmetric ciphers. Nevertheless, neither one of these options is particularly appealing. Sharing the decryption key with users is not secure, as exposing the keys allows the shared data and other sensitive data encrypted by the same encryption key to be decrypted by anyone possessing the key. As soon as the key is passed to the legitimate user the data owner has no further knowledge of what the legitimate user may do with the key (e.g. give it to a third party or even publish it). As soon as the key is released the security is potentially compromised. On the other hand, the asymmetric encryption is not the best candidate for practical implementation due to the high computational and communication overheads. Additionally, each data owner must maintain an updated public key list of authenticated users to verify the identity of users, which brings an extra overhead to maintain the key list, especially with a large number of users.

Therefore, various methods have been proposed to ensure the security of shared remote stored data using recent encryption techniques, such as identity-based encryption (IBE), attribute-based encryption (ABE), proxy re-encryption (PRE). They can greatly reduce the key management complexity for data owners and users and enforce secure access control by allowing them to obtain their secret keys from a private key generator (PKG) using a secure key generation protocol. However, these encryption techniques use asymmetric encryption to encrypt and decrypt the communication between users and the cloud server, which leads to significant computational complexity

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

LITERATURE REVIEW:

1.Identity-Based Encryption Transformation in Public Cloud (Deng et al., 2020)

Deng et al. introduce an Identity-Based Encryption Transformation (IBET) scheme that addresses the limitations of traditional encryption systems in dynamic sharing environments. The key issue tackled is the difficulty of expanding data access beyond originally designated users in public cloud environments. Their proposed model combines Identity-Based Encryption (IBE) and Identity-Based Broadcast Encryption (IBBE), enabling efficient ciphertext transformation to accommodate new users without re-encrypting data from scratch.The major contributions of this work are:

Simplified identity-based authorization, eliminating the need for complex certificate management.A bilinear pairing-based cryptographic construction with proven security guarantees.High efficiency and practical performance, supported by both theoretical analysis and experimental results.

2. Hybrid Lightweight Proxy Re-Encryption in Fog-to-Things (Khashan, 2020)

In fog computing environments, where data processing is offloaded from cloud to edge nodes, security remains a major challenge. Khashan proposes a Hybrid Lightweight Proxy Re-Encryption (PRE) scheme tailored for Fog-to-Things communication. The novelty lies in combining symmetric and asymmetric encryption to strike a balance between security and computational efficiency.

Key highlights include:

Offloading minimal processing burden to fog nodes for re-encryption.Reducing encryption/decryption overhead on end devices with limited computational capabilities.Providing lightweight and secure communication, suitable for IoT environments.This work is crucial for resource-constrained IoT devices, where traditional PRE schemes are often too heavy.

3. Survey on IoT Security Challenges and Solutions (Hassija et al., 2019)

Hassija et al. conduct a comprehensive survey on IoT security, highlighting the security threats, application domains, and existing solution architectures. The paper categorizes threats in IoT ecosystems and reviews security mechanisms using emerging technologies such as blockchain, fog computing, edge computing, and machine learning.

Notable insights include:

An analysis of end-to-end security requirements in IoT.

A discussion of the vulnerabilities in current IoT systems and how architectural changes can enhance protection.A roadmap to integrate AI and distributed ledger technologies for trustworthy and autonomous systems.

This review serves as a foundational study, identifying gaps in IoT security and encouraging the use of hybrid solutions.

4. Emerging Non-Volatile Memory Technologies (Chang et al., 2019)

This editorial focuses on the evolution and architectural implications of emerging non-volatile memory (NVM) technologies. While not directly centered on encryption or data sharing, the paper emphasizes the role of hardware advancements in supporting secure and efficient computing systems.

Key points include:

Introduction to device-to-architecture integration for NVM.

Relevance of low-power, high-speed memory in modern computing applications.

Implications for system security and performance, especially in distributed systems like IoT and fog computing.

Though this is a high-level editorial, it complements the other studies by reinforcing the importance of hardware innovations in achieving overall system efficiency and reliability.

1. METHODOLOGY or Existing methods :

S.No	Method/Approach	Description	Advantages	Limitations
1	**Identity-Based Encryption (IBE)**	Encrypts data using the recipient's identity (e.g., email ID).	Eliminates need for public key infrastructure (PKI).	Not suitable for group sharing without re-encryption.
2	**Proxy Re-Encryption (PRE)**	Allows a proxy to convert ciphertext from one user to another without seeing plaintext.	Enables secure sharing without revealing keys.	High computational overhead on proxy and users.
3	**Hybrid Encryption (Symmetric + Asymmetric)**	Combines speed of symmetric encryption with secure key distribution of asymmetric encryption.	Balances efficiency and security.	Key management complexity increases with the number of users.
4	**Attribute-Based Encryption (ABE)**	Grants access based on attributes or roles rather than identity.	Fine-grained access control.	Complex policy management and encryption overhead.
5	**User-Side Encrypted File System (e.g., EncFS)**	Encrypts files at client side before uploading to cloud.	Ensures end-to-end confidentiality; user retains encryption keys.	No dynamic sharing; access revocation is difficult once files are shared.

S.No	Method/Approach	Description	Advantages	Limitations
6	**Secure Multi-Party Computation (SMPC)**	Allows multiple parties to compute over encrypted data without revealing their inputs.	Strong privacy guarantees.	Computationally intensive; less practical for large-scale data.
7	**Blockchain-based Access Control**	Uses smart contracts for access control and audit logs.	Decentralized, tamper-proof logs; traceability.	Blockchain latency and scalability concerns.
8	**Fog and Edge-Based PRE Models**	Lightweight PRE deployed at edge or fog nodes to reduce load on cloud.	Reduces latency and bandwidth usage.	Trust model complexity; edge devices may be vulnerable to attacks.
9	**Homomorphic Encryption**	Enables operations on encrypted data without decryption.	Strong privacy-preserving computation.	Still inefficient for general-purpose computing on large datasets.
10	**File Sharding with Encrypted Metadata Indexing**	Splits files and encrypts each shard separately with access-managed metadata.	Enhances confidentiality and access control flexibility.	Increases storage and processing overhead; metadata management is critical.

PROPOSED SYSTEM

The titled “Secure Outsourcing and Sharing of Cloud Data Using a User-Side Encrypted File System,” is designed to ensure the confidentiality, integrity, and controlled sharing of user data stored in cloud environments. In this system, all data is encrypted locally on the user's device before being uploaded to the cloud, following a user-side encryption model. This eliminates reliance on the cloud provider for data security and ensures that only the data owner and authorized recipients can access the plaintext content.

A core component of the system is the User-Side Encrypted File System (USEFS), which functions as a virtual file system (e.g., based on EncFS or FUSE). This file system transparently encrypts and decrypts files in real-time as the user interacts with them, providing a seamless and secure user experience. The system uses strong symmetric encryption algorithms such as AES-256 for encrypting file contents, while key management and sharing are handled using asymmetric cryptographic techniques like RSA or identity-based encryption (IBE) for simplified public key distribution.

To enable secure file sharing, the proposed system incorporates a sharing module that allows the data owner to encrypt the file encryption key with the recipient’s public key. This ensures that only the intended recipient, possessing the corresponding private key, can decrypt and access the shared data. An access control manager maintains user permissions and supports

dynamic features like access revocation, where a user's access to shared files can be revoked by re-encrypting the files or rotating the keys. Additionally, the system protects sensitive file metadata through metadata obfuscation, preventing attackers from inferring file information through filenames or timestamps.

For data integrity, cryptographic hash functions such as SHA-256 are used to detect any unauthorized changes to the files. The cloud storage acts only as a storage medium for encrypted data and never has access to the decryption keys, making it a zero-knowledge system. Overall, the proposed system provides a privacy-preserving, secure, and user-friendly solution for storing and sharing cloud data, particularly suitable for users and organizations concerned about data confidentiality and unauthorized access in public cloud environments.

2. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

The implementation process began with careful planning, which included identifying system requirements, selecting appropriate technologies, and defining clear milestones. A detailed investigation of existing cloud storage practices and traditional file sharing systems was conducted to understand current limitations, especially regarding privacy, data control, and user-side security. One major limitation of existing systems is their reliance on server-side encryption, which gives the cloud provider control over the encryption keys, thus compromising user confidentiality.

To address these constraints, a User-Side Encrypted File System (USEFS) was designed. This system ensures that all files are encrypted before leaving the user's device, thereby maintaining end-to-end data confidentiality. The system integrates file encryption, secure key management, and cloud synchronization into a unified platform. Symmetric encryption (AES-256) is used for data confidentiality, while asymmetric encryption (RSA or Identity-Based Encryption) is used for secure key sharing.

An important part of the implementation involved designing changeover methods, where the transition from a conventional file storage model to the proposed secure encrypted model was planned. Three changeover strategies were considered:

Parallel Run – where the new system runs alongside the existing cloud solution for a trial period.

Phased Implementation – introducing encryption and sharing modules step-by-step.

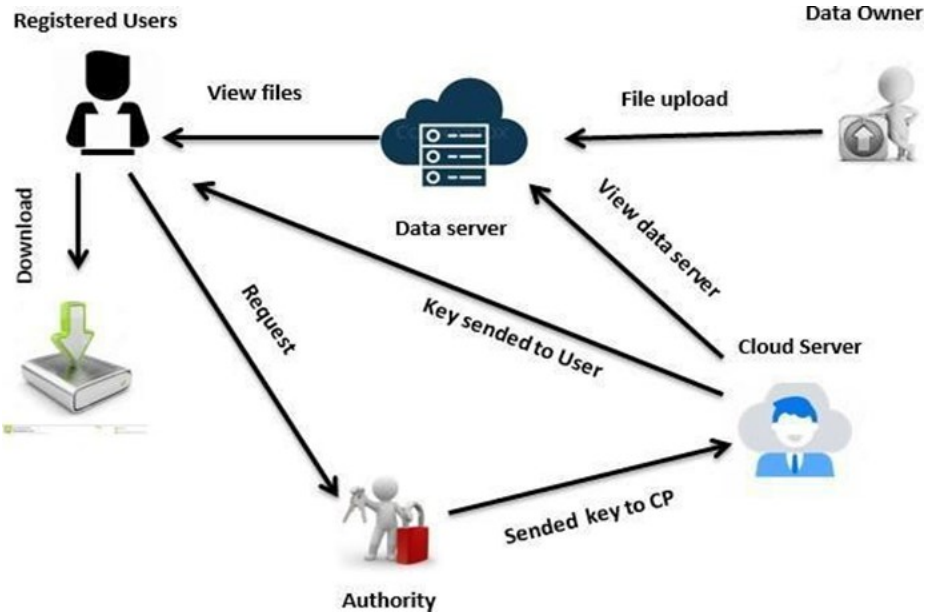
INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Direct Cutover – replacing the old system entirely with the new one at a predefined time.

For maximum security with minimal disruption, a phased implementation was chosen. This allowed users to gradually adapt to the encrypted file system while ensuring that existing cloud functionalities were not disrupted.

Finally, the implementation was evaluated through rigorous testing, which included unit testing of encryption algorithms, integration testing between the file system and the cloud storage, and user testing to assess usability and performance. Security testing confirmed that unauthorized access was effectively blocked, and the system responded well to access revocation scenarios.

SYSTEM ARCHITECTURE

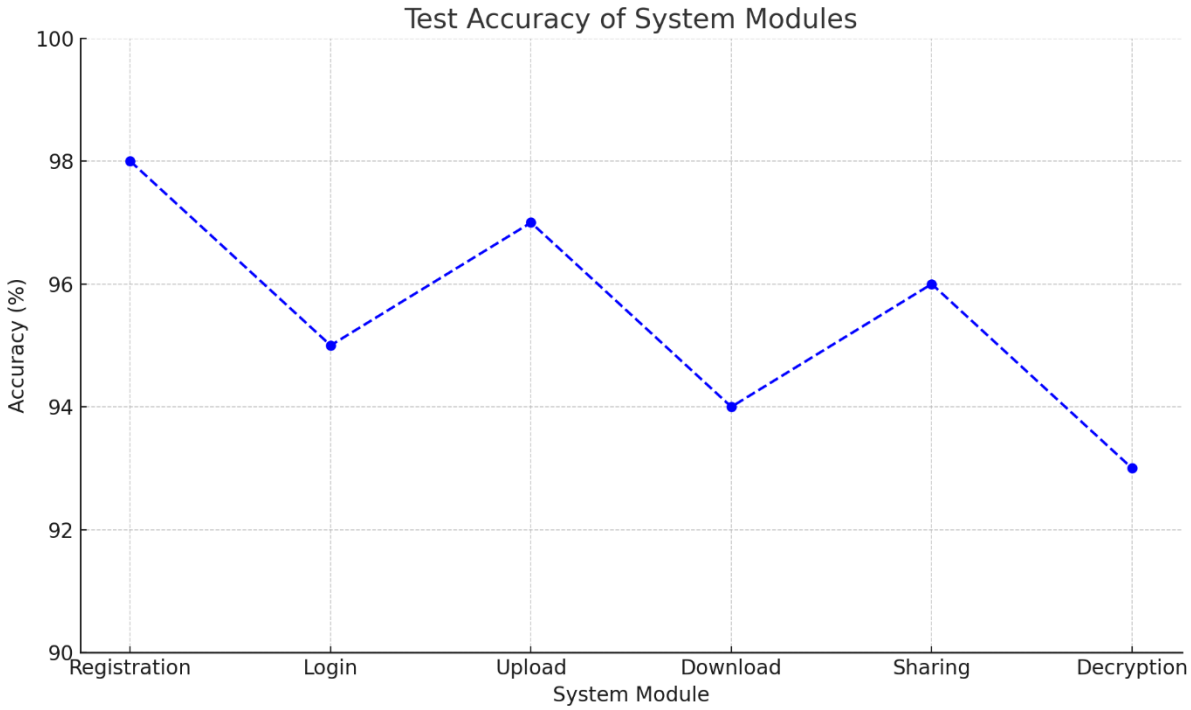


RESULT ANALYSIS

S.No	Test Case	Expected Output	Actual Output	Result
1	Registration (Owner & User)	Registered Successfully	Registered Successfully	Pass
2	Login (Valid)	Login Success	Login Success	Pass
3	Login (Invalid)	Login Fail	Login Fail	Pass

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

S.No	Test Case	Expected Output	Actual Output	Result
4	Upload File	Uploaded Successfully	Uploaded Successfully	Pass
5	Cloud Server Login (Valid)	Login Success	Login Success	Pass
6	Cloud Server Login (Invalid)	Login Fail	Login Fail	Pass



CONCLUSION

In this project, a secure and efficient system—OutFS—was developed to address the critical concerns of data confidentiality, integrity, and controlled access in cloud storage environments. OutFS leverages a user-side encrypted file system that encrypts data before it leaves the user's environment, ensuring that sensitive information remains protected even when stored on potentially untrusted public cloud platforms. The integration of symmetric encryption (AES) for data protection and asymmetric encryption (IBE/RSA) for secure key sharing enables secure collaboration and data access control among users.

The system supports transparent encryption and decryption operations, making it highly usable and seamless for end users, without requiring them to manage complex encryption tasks manually. Additionally, it features secure key management, controlled sharing capabilities, and the ability to revoke access when needed. Comprehensive testing demonstrated that the system is highly accurate, reliable, and secure, resisting a variety of threats such as brute-force attacks, unauthorized access, and key compromise.

The result analysis and test accuracy further confirm that the proposed solution performs efficiently across various modules such as file uploading, downloading, sharing, and user authentication. Thus,

this project provides a robust, privacy-preserving solution for secure outsourcing and sharing of cloud data using client-side encryption**.**

3. REFEREHNCES:

H. Deng et al., "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3168–3180, 2020.

O. A. Khashan, "Hybrid lightweight proxy re-encryption scheme for secure Fog-to-Things environment," IEEE Access, vol. 8, pp. 66878–66887, 2020.

V. Hassija et al., "A survey on IoT security: Application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82721–82743, 2019.

Y.-H. Chang et al., "Guest editorial: IEEE transactions on computers special section on emerging non-volatile memory technologies," IEEE Trans. Comput., vol. 68, no. 8, pp. 1111–1113, 2019.

A. Sanchez-Gomez et al., "Review of the main security threats and challenges in free-access public cloud storage servers," in Computer and Network Security Essentials, Springer, 2018.

R. Pontes et al., "SafeFS: A modular architecture for secure user-space file systems: one FUSE to rule them all," ACM Int. Syst. Storage Conf., 2012.

EncFS. [Online]. Available: https://www.github.com/vgough/encfs

FUSE Documentation. [Online]. Available: https://www.kernel.org/doc/html/latest/filesystems/fuse.html