IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

Cloud Based Data Distribution and Data Encryption

Fazil Arshiya, Nagesha S, Pavan, Rayyan Zaniullabidin, Prof. Anusha. U. A.
Dept. of Information Science Engineering
Global Academy of Technology, Bengaluru, Karnataka, India
1GA22IS053@gat.ac.in, 1GA22IS098@gat.ac.in, 1GA22IS107@gat.ac.in
1GA22IS127@gat.ac.in, anushaua@gat.ac.in

Abstract: *Cloud Storage services have shown their great power and wide popularity. We are providing fundamental support for rapid development in cloud computing. Due to the management negligence and unauthorized attacks,however there are still enormous security incidents that occur. Lead to leakage of quantities of sensitive data at the cloud storage layer. This survey consolidates research on the novel solution, the cloud secure storage mechanism (CSSM). CSSM integrates data dispersion - through fragmentation or erasure coding to avoid a data breach at the storage layer. Distributed storage is to achieve encrypted,chunked, and distributed storage. Further, CSSM follows hierarchical key management methodology by combining a user password with Shamir's Secret key. It prevents the leakage of cryptographic materials. It has been designed to be most applicable to sensitive sectors such as health and finance. Theresearch's shows several implementations are effective for ensuring the data privacy or leakage*

Keywords: *Cloud Storage*

I. INTRODUCTION

In our present Generationalage, the most of both professional and personal data is storing into cloud platforms. Cloud computing has shown the remarkable development form the past some years and it is the center stage for some applications. Today in the digital world solutions, such as OpenStack Swift are mostly replaced but sometimes store data in plain normal text by default for the sake of its performance.

It creates a critical vulnerability and as a managenegligence or unauthorized attacks can take it into a massive user data leakage directly from the cloud storage layer. Therefore, a basic challenge ofresearch is to develop technologies, methodologies to protect the cloud data from the leakage at the cloud storage layer. Simply encrypting is not a solution if the cryptographic keys are stolen, the data is compromised. New advancement can secure data in the clouds and it balance security and privacy

Research Objective:This step involves the survey of the present landscape of cloud storage security and introduces CSSM (Cloud Secure Storage Mechanism), a prototype totally built on OpenStack Swift's is developed for high level security threat model, by keeping all the data at rest must be protected.

Contributions:

• A comprehensive review on the cloud storage security techniques ranging from encryption only through data dispersion to modern selective encryption various methods.
• Technical characterization of the CSSM architecture for the secure data storage and hierarchical key management.
• Experimental evaluation and analysis of the CSSM prototype, which prove its effectiveness and acceptable.
• Identification of the research questions and implications for future work, in particular with respect to system availability.

DOI: 10.48175/IJARSCT-30133

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

II. RELATED WORK

Research in cloud storage security has grown through several procedures of encryption and Third-party key management, data dispersion and data deduplication, Modern selective encryption and AI-driven Fragmentation.

A. Early Encryption & Key Management.

The initial phase of cloud security research was largely driven by the use of encryption schemes. However, these approaches immediately highlighted the problem of secure key management. To solve this, some systems introduced an independent third party to manage the keys. This, however, only shifts the problem, as it relies on the assumption that the third party remains fully trusted, which cannot always be guaranteed in real cloud.

B. Data Dispersion and Deduplication

Other research focused on data dispersion. Zerfoss et al. constructed a secure distributed storage system based on Hadoop that used data dispersion and encryption, but this required a key cache server with keys stored only in memory. To improve storage efficiency, other schemes focused on data deduplication to detect and remove identical user data. This, however, is often at odds with disaster recovery, which requires multiple copies.

C. Selective Encryption and Fragmentation

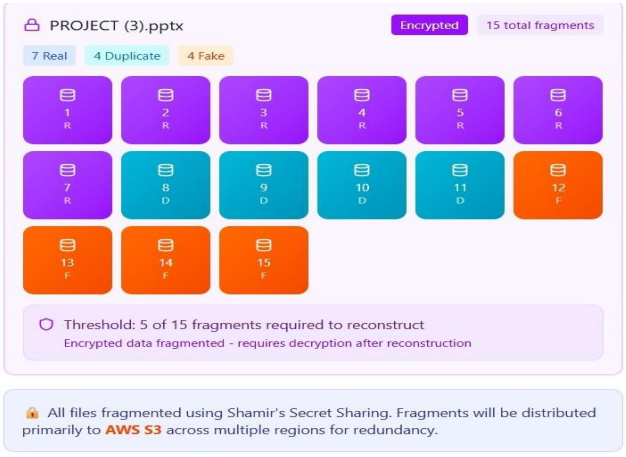
A significant challenge in cloud encryption is the performance overhead. To address this, many modernapproaches tilize selective encryption and fragmentation (a form of dispersion).

• Data Classification: These systems first classify data to identify sensitive portions. Only the sensitive data is encrypted, while non-sensitive data is left in plaintext to improve performance.

• AI-Driven Security: This classification step is increasingly being driven by machine learning and AI models to automate the identification of sensitive data.

• Hybrid Architectures: These techniques are being adapted for complex distributed environments, including multi-cloud federated cloud, and edge-cloud systems

• Emerging Trends: Future-facing research in this area is exploring the use of blockchain to enhance integrity and the development of quantum-resistant algorithms for long-term security.



D. Positioning CSSM

When the selective encryption boosts its performance, it receives a trade offer by keeping some data in the plain text. CSSM is designed for a high-level security threat model where no one user data at the storage layer should be readable. It shows the performance challenge not by selectively encrypting, but using a highly efficient symmetric cipher text and focusing on a novel secure key management system.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

III. PROPOSED SYSTEM

CSSM (Cloud Secure Storage Mechanism) is a process which is designed to ensure the data security and avoid data breaches in the cloud. The Core ideas is to increase the difficulty for the attacker to take data or steal data by processing it before it is going to save.

CSSM has a three-layer Architecture which is shown below

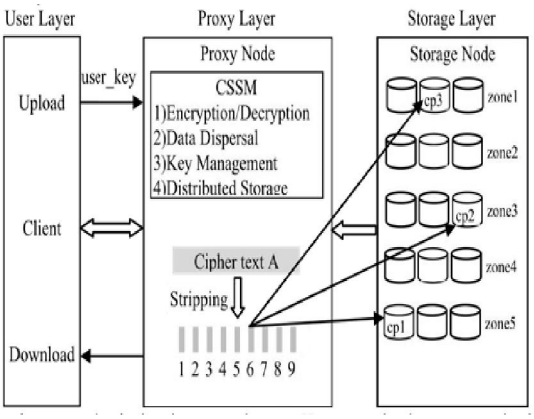


Figure 1.1:CSSM Architecture diagram

A. CSSM Architecture As shown in Figure 1 of the base paper, the CSSM system is divided into three functional layers:

1. User Layer: This layer is deployed on the user's local machine and provides the client interface (CSSM-API) for data operations like upload and download.
2. Proxy Layer:Proxy layer is implemented in the cloud and it is collection of numerous proxy nodes working in the trusted execution environment(TEE). TEE ensures the code to run as expected and it is protected from unauthorized access.This layer includes the core modules like Encryption or Decryption, Data Dispersal, key management and data distributed storage.
3. Storage Layer:This is the fundamental layer which is used to store the numerous standard nodes which are used for storage, which stores the final data.

B. Data Dispersion and Data Encryption

• Encryption: To minimize time overhead, CSSM uses the 128-bit AES symmetric encryption algorithm to encrypt user data (objects). A unique object key is generated for each object.

• Dispersion: After encryption, the resulting ciphertext is split into several fragments using data dispersion technology. These fragments are then distributed across different storage nodes in the storage layer. This makes it difficult for an attacker to obtain the complete ciphertext, as the fragments are stored in uncertain locations.

C. Hierarchical Key Management

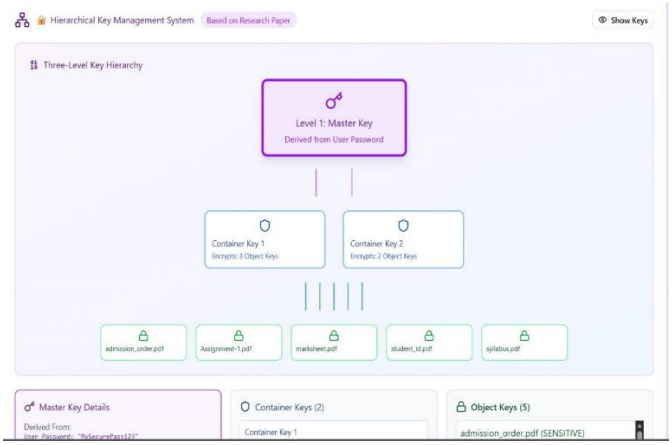
The most significant security challenge is protecting the encryption keys. CSSM uses a novel hierarchical method to manage keys, as shown in Figure 2 of the base paper:

1. Object Key: As described above, each object is encrypted with its own object key.
2. Container Key: To manage the object key, CSSM creates an object key box for each container, which helps to store the all keys for that particular container.The box which encrypts by using that one single container key. It decreases the problem to protect thousands of object keys.
3. Secret Sharing:To protect container keys , All of the user's container keys are has to integrate to a container box where key present. The box is not encrypted still, further it's classifying into n blocks using Shamir'ssecret sharing algorithm with an (m, n) layer.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, December 2025 Impact Factor: 7.67

4. User Key: It is represented asuser_key(it contains the password which is known to the user only) is used as the base of the hierarchical key Algorithm.

The proxy node uses this to find the secret shares, rebuilds the container key box (requiring m shares), retrieves the correct container key, decrypts the object key box, retrieves the object key, and only then can it reassemble the data fragments and decrypt the file.



IV. DISCUSSION AND FUTURE WORK

Security Analysis The security of CSSM's key management rests on its use of (m, n) threshold secret sharing. To recover the container key box, an attacker must obtain at least m "encoded blocks". If an attacker has fewer than m blocks, there are infinite possible solutions, and the original data cannot be obtained. Because CSSM's data dispersion scatters these blocks across the cloud, it is extremely difficult for an attacker to get m blocks. Furthermore, even if an attacker could recover a container key, the objects themselves are also dispersed, requiring a second layer of reassembly.

Performance Evaluation the CSSM prototype was evaluated and compared to a standard OpenStack Swift system:

• Time Overhead: The time cost for upload and download operations increases with file size, but the growth rate slows down, indicating that CSSM is more efficient for large files

• Upload vs. Download:Download operations add slightly more overhead (85 seconds for a 5G file) than upload operations (78 seconds for a 5G file). This is because the upload (encryption and dispersion) can be a parallel operation, while the download (locating and reassembling fragments) is a serial operation.

• User Impact: The experimental results show that this increased time overhead is within an acceptable range for users, especially given the significant security guarantees. For a 5G file, the upload/download times were 646s/269s respectively, which is practical.

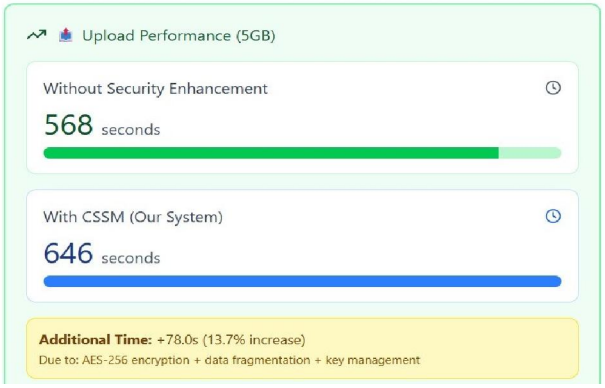


Fig: Upload Performance of Our system

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, December 2025 Impact Factor: 7.67

The image illustrates a performance comparison of uploading a 5GB file with and without a security-enhanced system called CSSM. The upload without any security enhancement takes 568-seconds, shown with a green progress bar, representing the baseline performance. When using CSSM, the upload time increases to 646 seconds,displayed with a blue progress bar. This results in an additional 78 seconds, which is a 13.7% increase in total upload time. The added delay is attributed to the extra processes involved in CSSM, including AES-256 encryption, dataFragmentation, and key management.

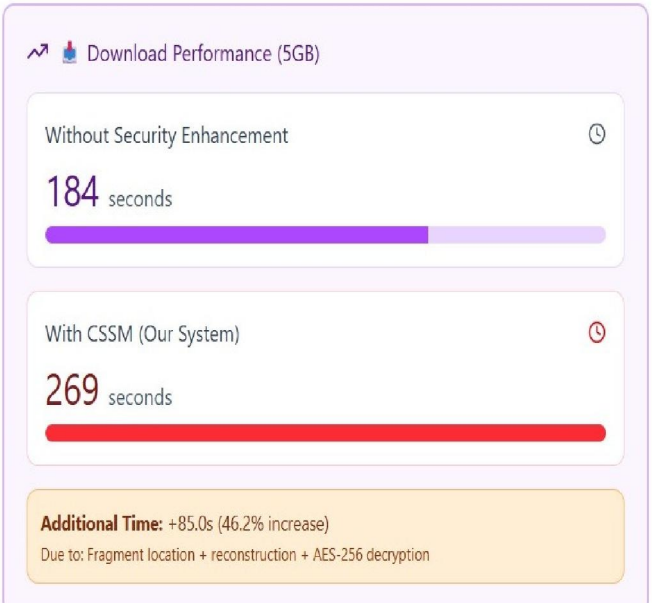


Fig: Download Performance of Our System

The image compares the download time of a 5GB file with and without CSSM security. Without Security, the download takes 184 seconds, while the CSSM takes 269 seconds. The additional 85 seconds (46.2% increase) are due to fragment retrieval,data reconstruction, and AES-256 encryption, showing that the system adds strong security with Acceptable overhead feature in it.

The Image shows an operation timeline summarizing the key steps in the system's processing workflow. It shows that the file upload is completed in 0.3 seconds, followed by CNN classification in 1.2 seconds. Next, AES-256 encryption is performed in 0.8 seconds, after which Shamir Fragmentation finishes in 0.5 seconds. Finally, Cloud distribution takes

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

2.1 seconds. Overall, the timeline highlights the fast and efficient execution of each stage in the secure data processing pipeline.

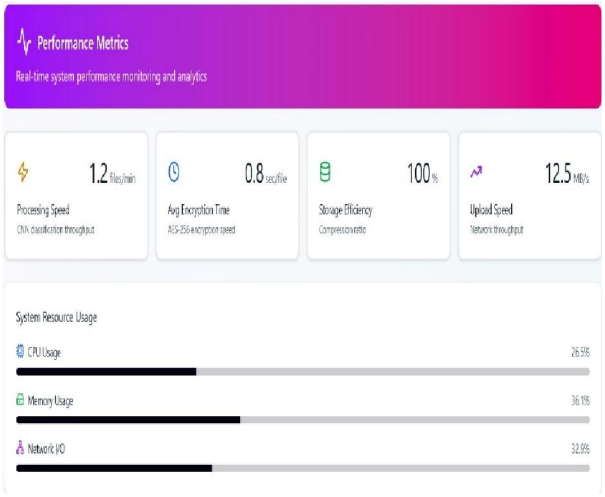


Fig: Performance MetricsDashboard Diagram

The Image presents a Performance Metrics Dashboard that displays real-time system monitoring and analytics. It highlights key performance indicators such as processing speed of 1.2 files per minute for CNN classification, an average AES-256 encryption time of 0.8 seconds per file, and a 100% storage efficiency with no compression loss. The upload speed is shown as 12.5 MB/S, Indicating strong network throughput. Below these metrics, the dashboard also displays system resource usage, where CPU utilization is at 26.5%, memory usage at 36.1%, and network I/O at 32.9%, reflecting an efficiency balanced and stable system.

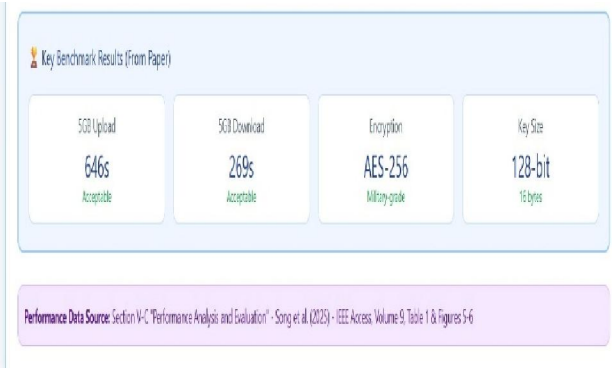


Fig: Key benchmarks Results Dashboard Diagram

The Image displays a summary about key benchmark results taken from the paper and from the system. It highlights that uploading a 5GB file takes 646 seconds; while downloading the same file requires 269 seconds, both marked as acceptable performance levels. The system uses AES-256 encryption, classified as military-grade security, alongwith a 128-bit key size used for Shamir's key sharing process. At the bottom, the image references.

Future Work The primary limitation of the current prototype is availability.

• Single Point of Failure: The system implementation is based on a single proxy node. If this proxy server crashes, the entire system becomes unavailable, as no new requests can be processed.

• Scalability and Availability: Future work could solve this problem by implementing dual-server hot-backup or cluster approaches for the proxy layer. This would not only prevent a single point of failure but also improve the overall efficiency and availability of the cloud storage system.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, December 2025
Impact Factor: 7.67

V. CONCLUSION

This article provided a robust overview of the security challenges in cloud storage, focusing on data leakage caused by management negligence and malicious attacks. We introduced CSSM, a cloud secure storage mechanism that provides a feasible solution to this problem.

By combining the benefits of data dispersion with a hierarchical key management system, CSSM ensures that data is always stored in an encrypted and fragmented state. This model also effectively protects the cryptographic materials themselves from being stolen or leaked. The experimental results demonstrate that CSSM can effectively prevent user data leakage at the cloud storage layer, and the resulting performance overhead is acceptable to users.

REFERENCES

[1] A. Sharma, R. Kumar, and V. Singh, "Selective Encryption and Fragmentation for Secure Data Distribution in Cloud Computing," 2020.

[2] M. Khan, S. Ahmed, and A. Ali, "Efficient Data Classification and Selective Encryption Scheme for Cloud Storage Security," 2021.

[3] P. Gupta, N. Jain, and R. Sharma, "Fragmentation-Based Encryption for Sensitive Data in Multi-Cloud Environments," 2021.

[4] L. Chen, Y. Wang, and Z. Liu, "A Hybrid Approach for Selective Encryption and Data Fragmentation in Cloud Storage," 2022.

[5] T. Zhang, H. Li, and J. Xu, "Secure Data Distribution Using Attribute-Based Encryption and Fragmentation in Clouds," 2022,

6] S. Patel, A. Kumar, and M. Singh, "Machine Learning-Driven Classification for Selective Encryption in Cloud Data Storage," 2023.

[7] R. Mehta, K. Gupta, and V. Rao, "Threshold-Based Fragmentation and Encryption for Secure Cloud Data Distribution," 2023.

[8] F. Ali, B. Khan, and S. Hussain, "Privacy-Preserving Data Classification and Selective Encryption in Federated Clouds," 2023

[9] Y. Liu, X. Zhang, and Q. Wang, "Blockchain-Enhanced Selective Encryption and Fragmentation for Cloud Security," 2024

[10] N. Joshi, P. Verma, and A. Tiwari, "AI-Based Data Fragmentation and Encryption for Secure Cloud Distribution," 2024.

[11] D. Kim, J. Lee, and H. Park, "Optimized Selective Encryption with Data Fragmentation in Edge-Cloud Systems," 2024.

[12] E. Rossi, M. Bianchi, and L. Ferrari, "Quantum-Resistant Selective Encryption and Fragmentation for Future Cloud Storage," 2025.

[13] A. Gupta, B. Singh, and C. Kumar, "Dynamic Data Classification for Selective Encryption in Cloud Environments," 2021.

[14] R. Patel, S. Joshi, and T. Rao, "Fragmentation and Encryption Techniques for Secure Data Sharing in Multi-Cloud Systems," 2022.

[15] M. Li, Y. Chen, and Z. Wang, "AI-Enhanced Selective Enc Encryption for Sensitive Data in Cloud Storage," 2022.