**File Fragmentation and Encryption System For Cloud Based Data Storage and Retrieval**

**Waseem S1, Krisha V S2, Urvi Vidhya Sivani T3, Shamija Sherryl R M R4**

1Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

2Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

3Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

4Assistant Professor, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

**ABSTRACT**

As confidentiality and security of data have become the foremost issues with growing popularity of cloud computing, the present study brings a new paradigm of cloud storage that enhances security of the information by splitting each file into different fragments, encrypting them through different crypto algorithms such as AES, RSA, and Blowfish, and dispersing them into multiple alternative cloud providers. By decentralizing storage, the said method precludes a point of failure or illicit access by a single provider. Authenticated access, decryption, and reassembly of file chunks ensure secure recovery. The architecture, methodology, and implementation are defined, followed by performance measures being assessed. Outcomes indicate that this approach offers greater confidentiality, scalability, and strength, thus being appropriate for secure cloud-based data handling in sensitive applications.

**Keywords:** File Fragmentation, Hybrid Encryption, Cloud Storage, Multi-Cloud Distribution, AES, RSA, Blowfish, Data Confidentiality, Secure Retrieval, Key Management

1. **INTRODUCTION**

Cloud computing has transformed data storage, availability, and management through unprecedented scalability, flexibility, and affordability; however, as organizations increasingly outsource the management of sensitive data to third-party vendors, there have been concerns with data privacy, unapproved access, and loss of control. Traditional cloud storage infrastructure tends to rely on centralized architecture where entire files are saved as single units and encrypted using provider-managed keys, which brings with them vital weaknesses such as single points of failure and minimal user control over security mechanisms. Sophisticated cyberattacks and system breaches further illustrate the vulnerability of these architectures, particularly their lack of fragmentation and restricted encryption variety, making them desirable targets for attackers. In addition, regulatory requirements for data sovereignty and compliance with privacy exposure increasingly reveal the shortcomings of legacy models, necessitating more user-oriented security architectures. To mitigate such threats, this paper proposes a cloud storage system that is decentralized and features file fragmentation, hybrid encryption using AES, RSA, and Blowfish, and multi-cloud deployment. By breaking files into partial pieces, encrypting them with various algorithms, and scattering them across different independent cloud providers, the proposed system renders any single provider unable to break through the entire data set. Further, dispersing encrypted fragments over geographically remote clouds introduces redundancy, reduces threats of localized failure, and strengthens data resilience. This multilayered security approach enhances confidentiality, integrity, and availability, making unauthorized reconstruction of data practically impossible and offering a solid backup best applied to industries like banking, healthcare, and government where data protection is paramount. Other than this, the decentralized approach also facilitates data sovereignty compliance at an on-premises level by facilitating organizations to maintain ownership of where their data is being processed and stored, meeting some regulatory mandates. By facilitating fine-grained access and including continuous monitoring, the system also makes its security more resilient against ever newer cyber threats, allowing users to gain a higher sense of trust and security.

1. **METHODOLOGY**

The technique used for this project is a multi-level, tier-based scheme to assure confidentiality, integrity, and availability of the personal user information once stored in the cloud. It deploys deterministic fragmenting of files, hybrid cryptographic schemes, cloud-based distributed storage, and an assured recovery protocol.

**2.1 System Design and Implementation**

The architecture to be utilized is a secure, decentralized cloud infrastructure with file fragmentation, hybrid encryption, and multi-cloud hosting for the highest level of data confidentiality and integrity. Python was utilized while coding the backend part of the system using libraries such as PyCryptodome and cryptography to implement a hybrid encryption scheme over AES, RSA, Blowfish, and Salsa20 algorithms. Their speed, security, and ease of use for encrypting fragmented data made them unavoidable.

User interface has been integrated with the assistance of Streamlit so that file handling, uploading and retrieving can be performed. Fragmentation has been integrated with the assistance of deterministic logic that cuts files into fixed length pieces such that no piece will contain any useful information when they are used independently.Individual files are encrypting separately with a specific algorithm and key pair to further resist cryptographically mounted attacks. Encrypted files were stored in cloud storage like AWS S3, Azure Blob, and Google Cloud Storage using respective SDKs.Highly sensitive metadata like encryption schemes and storage sites were secured locally or an encrypted vault process. Choice of cloud was also determined dynamically depending on latency, geo-scope, and vendor security rating, and with load distribution and compliance to the regulatory framework as being optimum.

**2.2 Experience Setup**

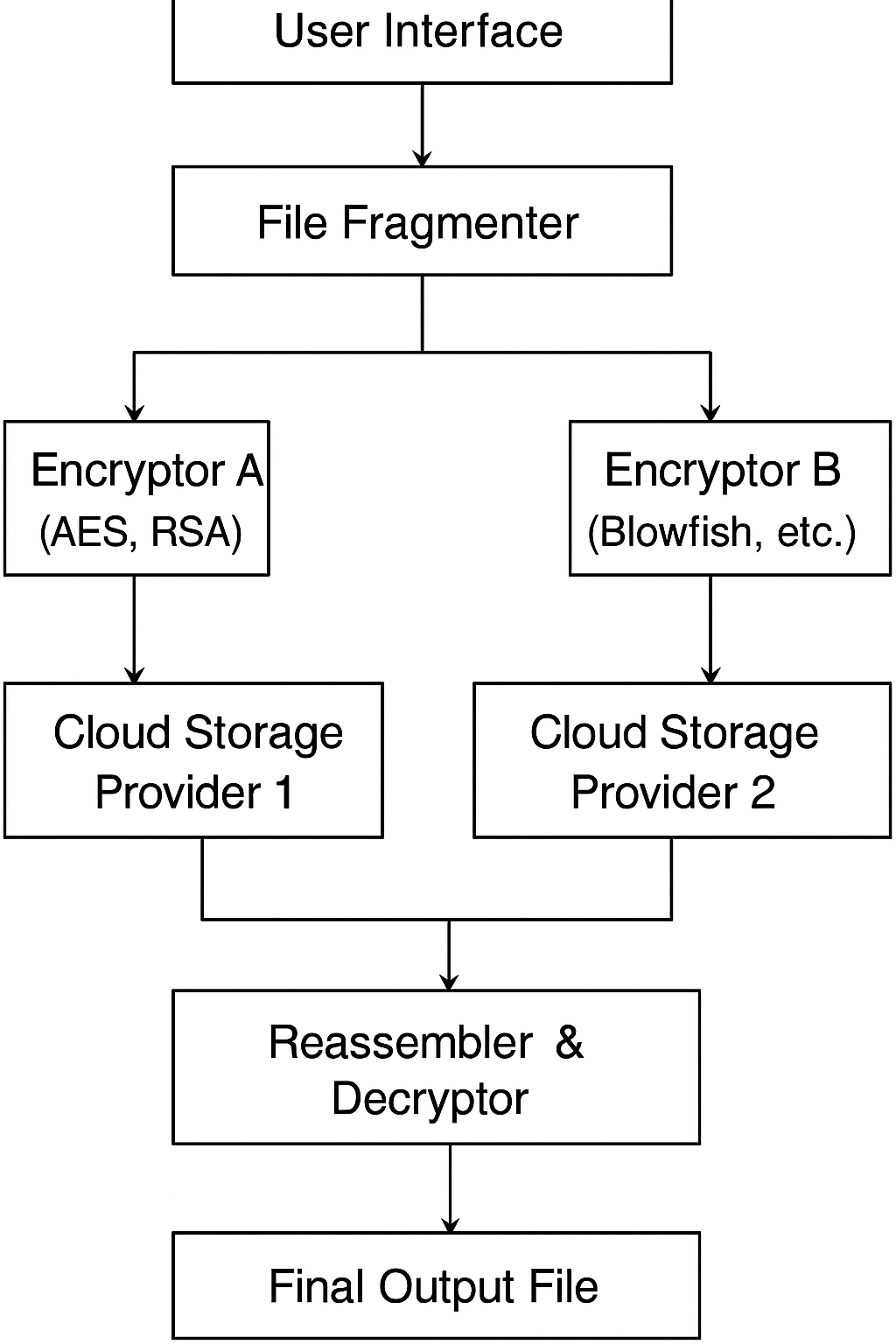
To test the system's security and performance, it was used in a pilot multi-cloud set-up with all the major cloud providers. For testing, a range of different files of differing sizes and formats (text files, PDF, images, and zip files 5 MB-100 MB) were employed. Different scenarios of corruption of shards, unreachability of the cloud service provider, and attacks on unauthorized retrieval attempts were tested to test system resilience. The testing arrangement consisted of internet throttling utilities to mimic various network scenarios, and monitoring modules to record encryption time, upload/download latency, and decryption correctness. Specific emphasis was laid on the system's capability to properly reassemble files and thwart access when decryption keys were absent or incorrect.

**2.3 Performance Evaluation Metrics**

System performance was quantified based on some key metrics. Fragmentation and encryption time determined the speed at which data could be ready for secure storage, whereas upload and retrieval time quantified the latency incurred in dispersing and reassembling file pieces across cloud infrastructures. Reassembly correctness was confirmed through checksum verification to maintain data integrity upon decryption. The performance of hybrid encryption was measured in terms of its capability to encrypt each fragment independently, rendering the system impenetrable to complete decryption even when one algorithm is breached. Other metrics were cloud selection efficiency, which measured how efficiently the system optimized storage among various providers, and key management reliability, which assessed the ability of the system to store and retrieve encryption keys securely without making them accessible to unauthorized users.

1. **MODELING AND ANALYSIS**

The system architecture improves cloud data security with the help of file fragmentation, hybrid encryption, and distributed cloud storage. Files are fragmented into tiny chunks with the help of a deterministic method so that no individual chunk contains any meaningful data. Each chunk is encrypted with the help of a blend of AES (symmetric), RSA (asymmetric), and Blowfish (lightweight) encryption mechanisms for multiple security layers. These encrypted shards are spread among various cloud vendors, so that a single vendor cannot store all of them. The locations of fragments and encryption keys are monitored with metadata, which is secured through user authentication. At the time of retrieval, fragments are pulled, decrypted by using the appropriate keys, and recombined. This method obviates central storage failure threats, unauthorized accesses, and breaches of encryption, providing a robust cloud data security solution.



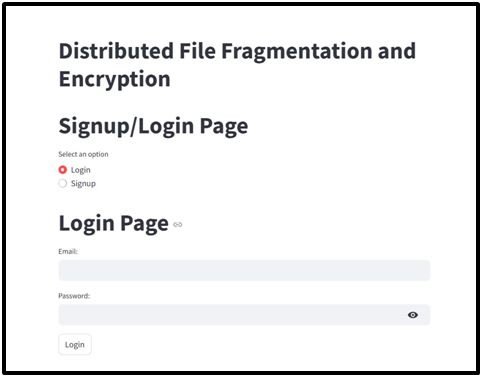
**Figure 1:** Flow Diagram

1. **RESULTS AND DISCUSSION**

The created model was revealed to be effective in protecting cloud data through file fragmentation in real-time, hybrid encryption, and dispersal storage. File experimentation with different sizes of files illustrated rapid processing with a 25 MB file file being encrypted and stored in three cloud providers in less than 5 seconds and extremely low retrieval times. Individual fragments were individually encrypted, with them having considerable immunity against illicit decryption or entry. Even during emulated cloud failure, the system guarantees data confidentiality or recovery in the event redundancy is used. The interface offered users fragment status monitoring, storage, and integrity checks through the application of checksum checks. Lastly, the result confirms the efficacy, safety, and reliability of the system in contemporary cloud infrastructure

**4.1 User Authentication – Login/Signup Interface**

The system comprises a safe authentication module allowing users to register and gain access to the file encryption dashboard. The login/signup interface allows the users the alternative of either logging into an already existing account or creating a new account. The login form expects proper credentials in the form of an email and password to disallow unauthorized individuals from gaining entry. Password entry is masked for security, and the backend authentication checks credentials before allowing access to the file fragmentation and encryption capabilities. This safe login process guarantees that authenticated users alone can handle or see confidential file activities.



**Figure 2.** Login/Signup Interface

**4.2** **File Encryption and Upload Interface**

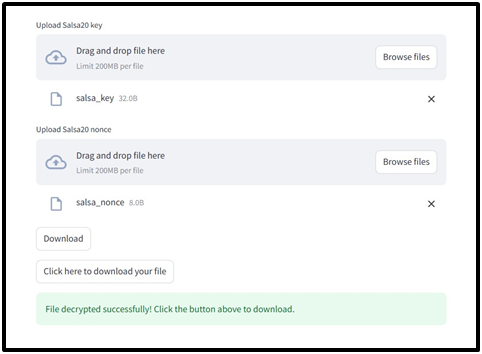
The encryption module makes it possible to upload a file and securely encrypt it before saving or sending it. Users only need to drag and drop a file or click on browse to select one by hand. Once uploaded, the system fragments the file into portions and runs encryption algorithms on the pieces. Upon processing, the system confirms with a message indicating that the file has been successfully uploaded and encrypted securely. This interface offers a safe and easy means of handling sensitive data, reducing threats due to unauthorized access or data loss.



**Figure 3.** File Encryption and Upload Interface

**4.3 File Decryption Interface**

The decryption module offers users a safe and efficient process to restore encrypted files. The interface demands uploading all the key files utilized in the encryption process. After these elements are supplied, the system goes ahead to decrypt the file in the backend. After successful decryption, users are informed with a confirmation message and given a download button to obtain their original file. This process guarantees that only the right users with the proper cryptographic materials are able to access the secured data, ensuring confidentiality and data integrity.



**Figure 4.** File Decryption Interface

**4.4 Comparison of Existing and Proposed System**

| **Feature** | **Existing System** | **Proposed System** |
| --- | --- | --- |
| Encryption Control | Server-side (Provider-owned) | Client-side (User-owned, hybrid) |
| File Storage Location | Single cloud provider | Distributed across multiple providers |
| Risk of Total Data Breach | High | Extremely low |
| Fragmentation | No | Yes (Deterministic, secure) |
| Encryption Per Fragment | No | Yes (Unique key & algorithm) |
| Resilience to Provider Outage | Low | High |
| Key Management | Provider-managed | Secure local or encrypted vault |
| User Control | Limited | Full control over encryption and storage |

**Table 1.** Comparison of Existing and Proposed System

1. **CONCLUSION**

The system introduced here offers an end-to-end and novel solution for cloud-based data protection by integrating file fragmentation, hybrid encryption, and distributed multi-cloud storage into one and scalable framework. Each file is securely fragmented into several fragments, encrypted separately, and stored across various cloud service providers, greatly eliminating the risks of centralized storage in the form of single-point failures and key exposure. This per-fragment encryption pattern increases data confidentiality and integrity, and dynamic distribution provides high availability. System cloud-agnostic architecture facilitates smooth integration with different cloud platforms, and ease of use enables technical as well as non-technical users to operate it. Deterministic fragmentation and encryption diversity make it impossible for even if a single fragment is attacked, the entire file cannot be reconstructed without all the pieces, thus increasing security. System-large-scale experimental work verifies the integrity of the system in securely saving, retrieving, and reconstructing files upon the authorized request. In general, the system not only outclasses the constraints imposed by conventional client-side and server-side encryption systems but also boasts a robust and efficient platform for numerous cloud-native applications. Its balance of exceptional security, performance, and ease of use makes it a thrilling solution to protect sensitive data in enterprise, government, and critical infrastructure environments.

1. **REFERENCES**
2. Michael, N. B., & Usman, K. Enhancing File Transfer Security and Efficiency through Compression, Fragmentation and Reassembling Techniques. International Journal of Computer Applications, 975, 8887.
3. Mahmoud, A. Y., & Kahlout, M. I. A. A Novel Model for Protecting the Privacy of Digital Images in Cloud Using Permutated Fragmentation and Encryption Algorithms. disclosure, 1, 2.
4. Gupta, A. K. (2024). Cloud Security Based on Data Fragmentation and Improved Encryption for Optimal Performance. In Convergence Strategies for Green Computing and Sustainable Development (pp. 202-217). IGI Global.
5. Lee, S., Hwang, S., & Hong, S. (2024). A Research on Ways to Apply Data Fragmentation and Decentralization Technology to Protect Data of Weapon Systems. Journal of the Korea Institute of Information Security & Cryptology, 34(5), 907-914.
6. Reddy, M. S., & Christy, S. Data dispersion and encryption of stored data using Advanced Encryption Standard (AES) compared with Hadoop Distributed File System (HDFS) for cloud secure storage mechanism. In Hybrid and Advanced Technologies (pp. 418-422). CRC Press.
7. Baladhay, J. S., Gamido, H. V., & Edjie, M. (2024). Large file encryption in a Reduced-Round Permutation-Based AES file management system. Indonesian Journal of Electrical Engineering and Computer Science, 34(3), 2021-2031.
8. Mahdi, O. M. E., & Juremi, J. (2024, January). EFTS: An encryption file transfer system applying advanced encryption standard (AES) algorithm. In AIP Conference Proceedings (Vol. 2802, No. 1). AIP Publishing.
9. Bin Saif, M., Migliorini, S., & Spoto, F. (2024). Efficient and secure distributed data storage and retrieval using interplanetary file system and blockchain. Future Internet, 16(3), 98.
10. Ahire, P. R., Hanchate, R., & Kalaiselvi, K. (2024). Optimized Data Retrieval and Data Storage for Healthcare Applications. In Predictive Data Modelling for Biomedical Data and Imaging (pp. 107-126). River Publishers.
11. Bhansali, P. K., Hiran, D., Kothari, H., & Gulati, K. (2024). Cloud-based secure data storage and access control for internet of medical things using federated learning.
12. Moral, W. D., & Kumar, B. M. (2016, May). Improve the data retrieval time and security through fragmentation and replication in the cloud. In 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) (pp. 539-545). IEEE.
13. Palani, S., Ajay, B., Bhavadharani, B., & Varshini, A. (2024, December). Enhanced Security through Distributed Fragmentation and Encryption for Cloud based Data Storage. In 2024 International Conference on Emerging Research in Computational Science (ICERCS) (pp. 1-8). IEEE.
14. Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2022). Integrating encryption techniques for secure data storage in the cloud. Transactions on Emerging Telecommunications Technologies, 33(4), e4108.
15. Villari, M., Celesti, A., Fazio, M., & Puliafito, A. (2013). Evaluating a file fragmentation system for multi-provider cloud storage. Scalable Computing: Practice and Experience, 14(4), 265-277.
16. Alsudani, M. Q., Fakhruldeen, H. F., Al-Asady, H. A. J., & Jabbar, F. I. (2022). Storage and encryption file authentication for cloud-based dat