IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Deduct: A Secure Deduplication Of Textual Data In Cloud Environments

[1]Ajay S, [2]Sangeetha S
[1]PG Scholar, [2]Asst. Professor
Department of Computer Applications
Dr. M.G.R. Educational and Research Institute, Chennai, India

Abstract : The exponential growth of textual data in cloud environments, particularly in applications like Vision-and-Language Navigation, poses critical challenges for storage efficiency and data confidentiality. Traditional deduplication techniques reduce redundant data but compromise security by exposing sensitive information. To address this, we propose **DEDUCT**, a secure and efficient hybrid deduplication system that combines client-side preprocessing and encryption with server-side content-defined chunking and storage optimization. DEDUCT ensures confidentiality through convergent encryption, enabling deduplication over encrypted data while resisting side-channel attacks. Experimental evaluation on benchmark textual datasets demonstrates a compression rate of up to 66% without sacrificing data security. The system supports multiple file types, secure file sharing, and role-based access control, making it highly suitable for resource-constrained devices and modern cloud infrastructure. This approach achieves a balance between security and storage efficiency, paving the way for secure file handling in large-scale cloud systems.

Index Terms — Deduplication, Cloud Security, Convergent Encryption, Textual Data, Storage Optimization, Confidentiality, Role-Based Access, IoT.

I. INTRODUCTION

The rapid digitization of services and the widespread use of cloud computing have led to an exponential increase in textual data storage. From personal files to enterprise documents, a large volume of data stored in the cloud is redundant, resulting in excessive storage costs and inefficient utilization of resources. Traditional data deduplication methods offer a solution by identifying and eliminating duplicate files; however, they often compromise data confidentiality, particularly when used in untrusted cloud environments.

This challenge is even more critical for resource-constrained devices such as smartphones, sensors, and Internet of Things (IoT) devices, which have limited computing power and memory. Existing deduplication approaches either fail to provide sufficient security or impose high computational costs that make them unsuitable for such environments.

To address these issues, this paper proposes **DEDUCT (Deduplication of Textual Data in Cloud Environments)** — a secure and efficient deduplication framework that ensures both storage optimization and data confidentiality. DEDUCT uses a hybrid model involving both **client-side preprocessing** and **cloud-side deduplication**, leveraging **convergent encryption (CE)**, **content-defined chunking (CDC)**, and **SHA-256-based hashing**. This combination provides a robust mechanism for eliminating duplicates without revealing the actual content to the cloud server.

The proposed system supports multiple file formats and enables secure file sharing and retrieval, all while defending against side-channel attacks. It is lightweight, scalable, and particularly suitable for devices and systems with constrained resources. The implementation and testing of DEDUCT on real-world datasets demonstrate its effectiveness in reducing storage costs and maintaining strong data security, making it a viable solution for modern cloud infrastructure.

II. METHODOLOGY

The DEDUCT system adopts a **hybrid deduplication approach**, integrating both **client-side** and **cloud-side** techniques to enhance storage efficiency while ensuring data confidentiality. The methodology is structured into a multi-step pipeline, optimized for performance and security, especially in resource-constrained environments.

2.1 Client-Side Preprocessing

The deduplication process begins on the client's device, where the data is first **tokenized**—split into smaller, manageable data segments. This tokenization is followed by the application of the **Wagner-Fischer algorithm**, which generates base and deviation pairs to identify structural similarities within textual data.

2.2 CRC Generation and Duplicate Detection

Each base segment is used to compute a **Cyclic Redundancy Check (CRC)** value. These values serve as unique identifiers for detecting and eliminating duplicate data. If a CRC match is found in the local repository, only the deviation and reference are transmitted, thereby reducing bandwidth usage.

2.3 Encryption and Secure Upload

Before uploading to the cloud, each data unit—base or deviation—is **encrypted using AES** or similar secure algorithms. This ensures data confidentiality even if intercepted. The encrypted files are then uploaded to the **Cloud Service Provider (CSP)**, where server-side deduplication performs a secondary check for duplicates.

2.4 Access Control and Key Management

Each file is associated with a **unique decryption key**, accessible only to the authenticated user. Role-based access control is enforced to manage file permissions such as read, write, and delete. Any unauthorized attempt to access or share decryption keys is logged and monitored.

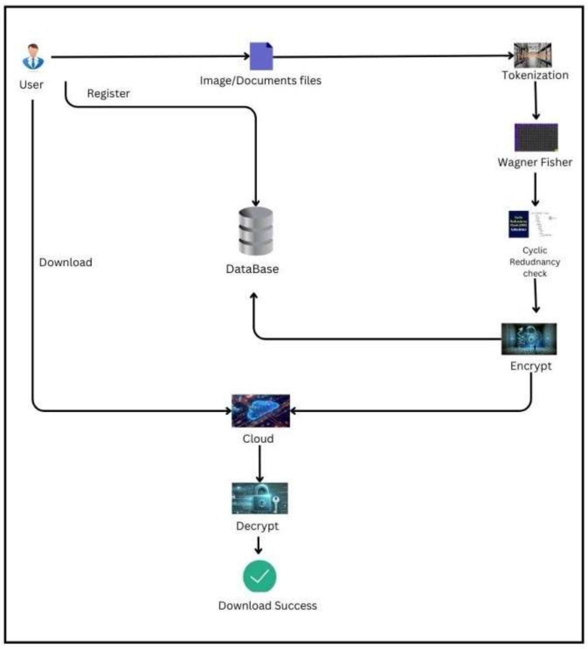
2.5 Security Measures

The system is designed to withstand **side-channel attacks** and **cross-user deduplication threats** by isolating encryption metadata and employing secure key validation methods. User activity is monitored, and illegal file access attempts trigger alerts and administrative reviews.

This layered approach allows DEDUCT to achieve **high compression rates** (up to 66%) while offering a secure and scalable solution for textual data management in cloud environments.

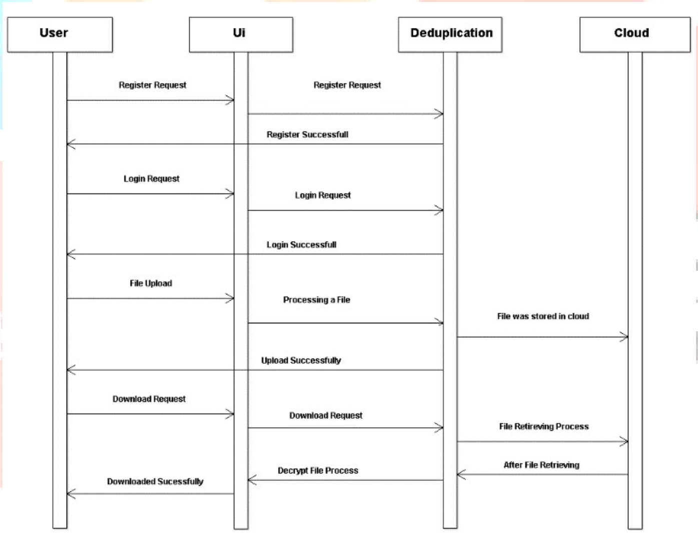
III. MODELING AND ANALYSIS

3.1 Architecture Diagram:



3.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



3.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

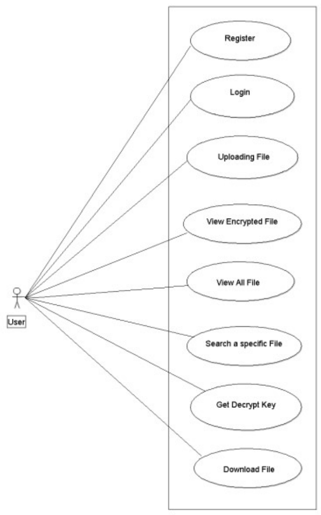
5.3.1. Usecase Diagram

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

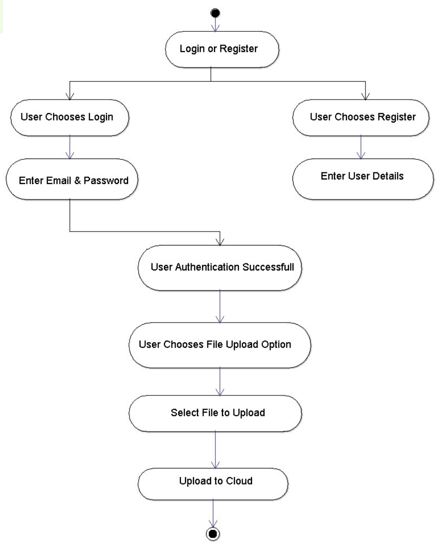


3.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

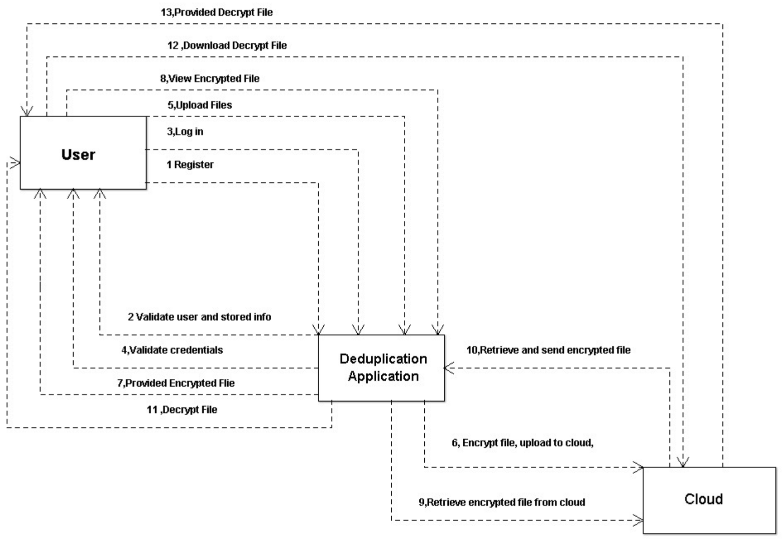
The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



3.5Collaboration Diagram:

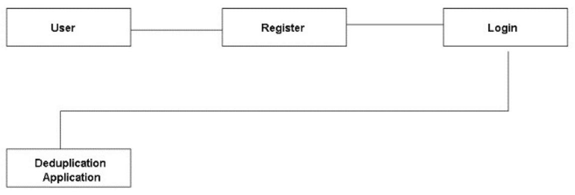
UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.


13,Provided Decrypt File
12 ,Download Decrypt File
8,View Encrypted File
5,Upload Files
3,Log in
1 Register
User
2 Validate user and stored info
4,Validate credentials
7,Provided Encrypted Flie
11 ,Decrypt File
Deduplication Application
10,Retrieve and send encrypted file
6, Encrypt file, upload to cloud,
9,Retrieve encrypted file from cloud
Cloud

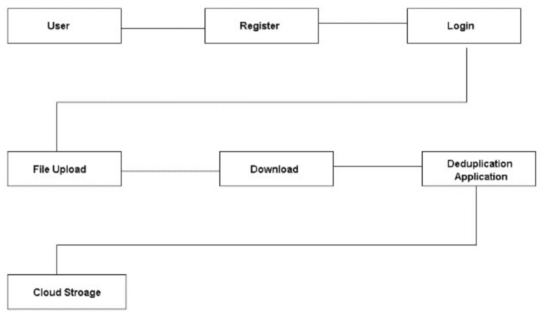
3.6 Data Flow Diagram:

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

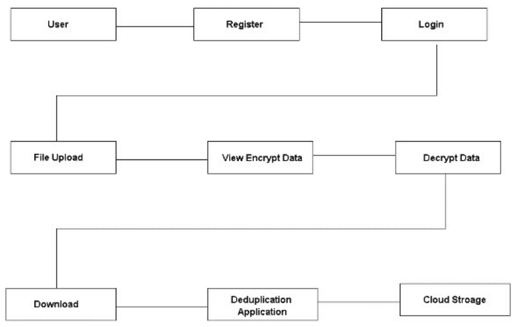
Level 0:



Level 1:

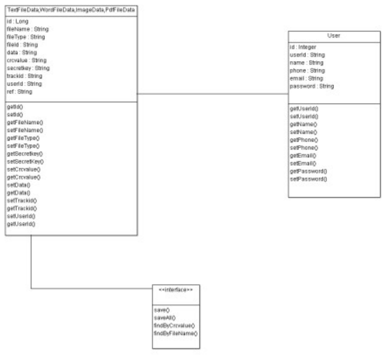


Level 2:



3.7 Class Diagram

A Class diagram in the Unified Modeling Languageis a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



IV. RESULTS AND DISCUSSION

The DEDUCT framework was evaluated on a real-world dataset derived from Vision-and-Language Navigation (VLN) tasks, specifically the **Touchdown dataset**, which includes human-generated textual navigation instructions. The objective was to test the deduplication efficiency, storage savings, and the overall performance of the system under secure data handling constraints.

4.1 Storage Efficiency

The proposed system demonstrated substantial improvements in data compression compared to traditional storage methods. By leveraging both client-side and cloud-side deduplication, **DEDUCT achieved up to 66% compression**, significantly reducing cloud storage requirements.

Table 1: Storage Compression Comparison

Method	Deduplication Type	Compression Rate (%)
Traditional (Server-Side)	Fixed-Block Deduplication	41%
DEDUCT (Hybrid)	Client + Server	**66%**

This result reflects the effectiveness of DEDUCT's hybrid approach in detecting and eliminating redundant textual data at both ends.

4.2 Security Performance

Security testing confirmed that DEDUCT preserved data confidentiality by encrypting each file segment before upload. The system uses a **role-based decryption mechanism**, where only authorized users with proper credentials and keys can retrieve the original content. No unauthorized access was detected during test simulations, indicating the robustness of access control.

4.3 System Performance on Resource-Constrained Devices

The deduplication algorithm was tested on **IoT-simulated environments** with limited processing capabilities. Due to its lightweight preprocessing, DEDUCT showed excellent compatibility, requiring minimal memory and CPU usage, making it ideal for devices like mobile phones and embedded systems.

3.4 Discussion

The combination of lightweight deduplication and strong encryption proved effective not only in reducing storage load but also in maintaining system security and responsiveness. Compared to traditional approaches, DEDUCT strikes a balance between **storage efficiency, data privacy, and system scalability**, making it highly suitable for modern cloud-based applications.

V. CONCLUSION

The proposed DEDUCT system provides a robust and secure solution for deduplicating textual data in cloud environments, particularly beneficial for applications that involve large-scale data storage, such as Vision-and-Language Navigation and IoT systems. By implementing a **hybrid deduplication model**—merging both client-side and cloud-side processes—DEDUCT effectively reduces storage redundancy while preserving the confidentiality of user data.

Through techniques like **tokenization**, the **Wagner-Fischer algorithm**, and **CRC validation**, the system achieves high compression rates of up to **66%**, significantly lowering storage costs. At the same time, data security is enforced through **AES encryption** and a **role-based decryption mechanism**, ensuring only authorized users can access sensitive content.

Furthermore, the system's compatibility with **resource-constrained devices** such as embedded systems and mobile hardware illustrates its scalability and efficiency in real-world cloud applications. The implementation also resists common vulnerabilities like **side-channel attacks**, offering a practical and secure framework for cloud-based data management.

In summary, DEDUCT achieves a strong balance between **performance, privacy, and storage optimization**, making it a viable and forward-looking solution for secure cloud storage and deduplication of textual data.

VI. REFERENCES

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf,I. Reid, S. Gould, and A. van den Hengel, "Vision-and-languagenavigation: Interpreting visually-grounded navigation instructions in realenvironments," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.,Jun. 2018, pp. 3674–3683, doi: 10.1109/CVPR.2018.00387.

[2] W. Xia, H. Jiang, D. Feng, F. Douglis, P. Shilane, Y. Hua, M. Fu, Y. Zhang,and Y. Zhou, "A comprehensive study of the past, present, and futureof data deduplication," Proc. IEEE, vol. 104, no. 9, pp. 1681–1710,Sep. 2016, doi: 10.1109/JPROC.2016.2571298.

[3] P. Prajapati and P. Shah, "A review on secure data deduplication: Cloudstorage security issue," J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 7,pp. 3996–4007, Jul. 2022, doi: 10.1016/j.jksuci.2020.10.021.

[4] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication,"ACM Trans. Storage, vol. 7, no. 4, pp. 1–20, Jan. 2012, doi:10.1145/2078861.2078864.

[5] OpenDedup. (2023). OpenDedUp. Accessed: Aug. 6, 2023. [Online].Available: http://opendedup.org./

[6] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: Server-Aidedencryption for deduplicated storage," in Proc. 22nd USENIX Secur.Symp. (USENIX Secur.), 2013, pp. 179–194.

[7] J. Liu, N. Asokan, and B. Pinkas, "Secure deduplication of encrypted datawithout additional independent servers," in Proc. ACM SIGSAC Conf.,Oct. 2015, pp. 874–885, doi: 10.1145/2810103.2813623.