International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.1 (2025) pp. 249-260
http://www.ijcesen.com

ISSN:2149-9144

Research Article

Multi-layer access control for cloud data using improved DBSCAN, AES, homomorphic encryption, and RMAAC for text, image, and video

A. Jeneba Mary[1*], K. Kuppusamy[2], A. Senthilrajan[3]

[1]Research Scholar, Department of Computational Logistics, Alagappa University, Karaikudi, Tamilnadu, India
* **Corresponding Author Email:** jenebamary@gmail.com - **ORCID:** 0009-0008-0404-4197

[2]Formerly Professor & Head(i/c), Department of Computational Logistics, Alagappa University, Karaikudi, Tamilnadu, India
Email: ksamyk@alagappauniversity.ac.in - **ORCID:** 0009-0002-8182-1139

[3]Professor Department of Computational Logistics, Alagappa University, Karaikudi, Tamilnadu, India
Email: senthilrajana@alagappauniversity.ac.in - **ORCID:** 0000-0003-4265-7097

Article Info:

DOI:10.22399/ijcesen.837
Received : 01 October-2024
Accepted : 30 December 2024

Keywords :

DBSCAN,
CLOUD,
RMAAC,
AES,
Security,
Homomorphic Encryption.

Abstract:

With the growing demand for cloud storage solutions that can handle diverse data types such as text, images, and videos, ensuring robust access control and security becomes important. This paper proposes a novel multi-layer access control framework for cloud environments, incorporating advanced Clustering Filtering Techniques with an Improved DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm to enhance search efficiency across multiple data layers. This clustering approach enables quick and accurate retrieval of text, image, and video content by efficiently organizing data based on similarity. To ensure data privacy and security, employs a hybrid encryption approach, combining Advanced Encryption Standard (AES) for data at rest and Homomorphic Encryption (HE)for data in use, allowing secure data manipulation without compromising confidentiality. The access control mechanism is further strengthened by introducing a Role-based Multi-Attribute Access Control (RMAAC) model, which grants permissions based on a user's role, attributes, and the sensitivity level of the data being accessed. This fine-grained control restricts unauthorized access while supporting flexible policies for different data types. Simulation results demonstrate that the proposed framework significantly improves data retrieval speed, security, and clustering performance, making it an effective solution for cloud storage systems handling diverse media formats.

1. Introduction

The exponential growth of digital data and the widespread adoption of cloud computing have revolutionized how individuals and organizations store, process, and access information. As data increasingly resides on cloud platforms, ensuring secure and efficient access control mechanisms has become a critical concern [1,2]. Cloud computing offers unparalleled benefits such as scalability, cost efficiency, and ease of access. However, these advantages are accompanied by significant challenges related to data security, privacy, and unauthorized access, especially for sensitive content such as text documents, images, and videos. Access control is a foundational element of cloud security [3]. It ensures that only authorized users can access specific data, preventing potential misuse or unauthorized exposure. Traditional single-layer access control mechanisms, while effective in some cases, often fall short in addressing the diverse and complex needs of modern cloud environments [4]. The varying sensitivity levels of textual, visual, and multimedia data demand tailored approaches to ensure that the integrity and confidentiality of each type of data are maintained. This calls for a robust and adaptable multi-layer access control system that can manage and secure data across different content types [5,6].

A multi-layer access control framework integrates various security layers, leveraging authentication, authorization, encryption, and auditing mechanisms to create a comprehensive protection strategy. For textual data, fine-grained access controls ensure that

only relevant users can view or edit documents [7]. For images, additional safeguards such as watermarking or metadata-based restrictions can be implemented. For video data, advanced techniques like streaming-specific access policies and encryption during transmission offer enhanced security. Such a framework not only reinforces data security but also ensures compliance with data protection regulations and industry standards [8].One of the key drivers for adopting multi-layer access control systems is the growing reliance on cloud platforms for storing and processing multimedia data. Images and videos are particularly vulnerable to misuse, including unauthorized sharing or tampering, which can have significant repercussions for both individuals and organizations [9]. Similarly, textual data such as confidential reports or sensitive legal documents require stringent access control to prevent data breaches. A unified access control solution capable of addressing the unique requirements of these data types provides a significant advantage in safeguarding digital assets [10,11].

The implementation of multi-layer access control for text, image, and video data requires a combination of advanced technologies, including cryptographic algorithms, role-based access control (RBAC), and artificial intelligence (AI)-driven monitoring systems. These technologies work in tandem to ensure that data is accessible only to the intended users under predefined conditions, thereby minimizing the risk of data breaches [12,13]. Furthermore, continuous monitoring and adaptive security policies enable the system to respond to evolving threats dynamically [14,15]. This paper presents a comprehensive exploration of multi-layer access control mechanisms for cloud data, with a specific focus on text, image, and video content. It highlights the challenges inherent in managing access control across these diverse data types and proposes innovative solutions to address these issues. By examining real-world use cases and evaluating the performance of proposed strategies, this study aims to contribute to the development of secure and efficient cloud environments. Ultimately, multi-layer access control frameworks not only enhance data security but also build trust among users, fostering greater adoption of cloud technologies in sensitive and high-stakes domains.

The rapid adoption of cloud storage solutions has significantly transformed how organizations store and manage diverse data types, including text, images, and videos. While cloud platforms offer scalability and convenience, they also introduce several challenges related to data security, privacy, and access control. Sensitive data uploaded to the cloud is vulnerable to unauthorized access, leading to concerns about data leakage, manipulation, and privacy breaches. Traditional encryption methods, such as AES, protect data at rest, but they do not address the issue of data in use or the need for efficient retrieval of specific data types. Furthermore, managing access control policies for multiple data formats and users with varying privileges is a complex task. There is a need for an effective solution that can securely manage data across different formats, enhance search efficiency, and ensure fine-grained, role-based access control without compromising the performance of the cloud storage system.

The contributions of this paper are manifested below,

- This work proposes a multi-layer access control framework for cloud environments, integrating advanced Clustering Filtering Techniques with an Improved DBSCAN algorithm to enhance search efficiency across various data layers, such as text, images, and videos.
- This work introduces a hybrid encryption approach combining AES for data at rest and HE for data in use, ensuring secure data manipulation without compromising confidentiality in cloud storage systems.
- This work presents a RMAAC model that provides fine-grained access control by granting permissions based on a user's role, attributes, and data sensitivity, effectively restricting unauthorized access.
- This work demonstrates through simulations that the proposed framework significantly improves data retrieval speed, security, and clustering performance, making it a highly effective solution for cloud storage systems dealing with diverse media formats.

This paper is further divided into the following sections. The part 2 presents both related works and problem statement. The suggested method is implemented and illustrated in the part 3. The result and discussion are then presented in the part 4, followed by the conclusion in the part 5.

2. Related Works

Anju and Shreelekshmi [16] presented an efficient and scalable secure content-based image retrieval scheme for the cloud. The image owner extracts MPEG-7 visual descriptors from images, clusters them for indexing, and encrypts the image features and cluster centers using Asymmetric Scalar-Product Preserving Encryption. These encrypted descriptors and images are offloaded to the cloud to enable secure retrieval. Additionally, a copy-deterrence mechanism is introduced to detect

untrustworthy query users, ensuring faster retrieval and improved perceptual quality of watermarked images, with comparable extraction accuracy even when compromised.

Ramachandra *et al.* [17]introduced the Triple Data Encryption Standard (TDES) methodology, enhancing security by increasing key sizes in the Data Encryption Standard (DES). Experimental results demonstrate that TDES effectively secures healthcare data in the Cloud, showing lower encryption and decryption times compared to the existing Intelligent Framework for Healthcare Data Security (IFHDS).

Rafique*et al.* [18]proposed CryptDICE, a distributed data protection system designed to address data security and privacy concerns in cloud storage. It provides built-in support for various data encryption schemes, accessible via annotations based on application-specific requirements. CryptDICE enables users to make appropriate trade-offs between performance, security, and storage efficiency at different levels of data granularity.

Huang*et al.* [19]proposed a secure image retrieval scheme that addresses privacy concerns in outsourcing image datasets to public clouds. The scheme enhances search accuracy by extracting image features using fine-tuned convolutional neural networks, which are then encrypted using the secure k-Nearest Neighbor algorithm. To reduce computational costs and improve search speed, cloud servers locally build a secure hierarchical index graph with encrypted image features. The proposed scheme allows for parallel index construction and updates, minimizing the image owners' cost and time compared to existing methods.

Xu*et al.* [20] used the Hamming embedding algorithm to generate binary signatures for image descriptors. A frequency histogram, combined with binary signatures, enhances the accuracy of image feature representation. Visual words are selected via random sampling, followed by min-Hashing on the selected words to generate a secure index. This approach balances security, accuracy, and efficiency in large-scale image retrieval. Security analysis and experiments validate the method's effectiveness.

Li*et al.* [21]developed the privacy-preserving content-based image retrieval scheme that combines asymmetric scalar-product-preserving encryption (ASPE) and (HE) to ensure key confidentiality and protect against attacks from data owners, cloud servers, and users. The scheme is designed under a strong threat model, assuming that all entities involved are semi-trusted. It safeguards the confidentiality of keys and query privacy, while also incorporating a lightweight verification process to detect fake search queries.

Chai*et al.* [22] proposed thumbnail-preserving encryption (TPE) technique uses a genetic algorithm, where image pixels in sub-blocks are scrambled and diffused at the bit level through crossover and mutation operators. New operators, Mutation Compensation and Mutation Failure, enhance the TPE to maintain the thumbnail of the original image. A color histogram-based retrieval algorithm is introduced to improve search accuracy using Bhattacharyya distance, and simulations confirm the security and effectiveness of the scheme.

Shen*et al.* [23]presented a secure content-based image retrieval (CBIR) scheme, supporting multiple image owners with privacy protection (MIPP). To address privacy concerns, image features are encrypted using a secure multi-party computation technique, allowing each owner to encrypt their features with their own keys. This enables efficient retrieval across images from multiple sources while preserving individual image owners' privacy. Additionally, a new similarity measurement method is introduced to prevent revealing image similarity information to the cloud, ensuring secure image retrieval.

Xu*et al.* [24]used a privacy-preserving content-based image retrieval (CBIR) method using orthogonal decomposition to protect sensitive image data. The image is split into two components, with encryption and feature extraction performed separately. This allows the cloud server to directly extract features from the encrypted image and compare them with the queried image features, enabling secure retrieval. Unlike other methods, the proposed approach does not rely on specific encryption algorithms, making it more versatile and suitable for various applications and scenarios.

Xia*et al.* [25] presented a privacy-preserving image retrieval scheme where images are encrypted, yet similar images to a query can be efficiently retrieved. The image content is protected using multiple encryption techniques, including big-block and pixel permutations, along with poly alphabetic cipher, which enhances security without degrading retrieval accuracy. Secure Local Binary Pattern (LBP) features are directly extracted from the encrypted images, eliminating the need for communication with the image owner. The LBP features generate a feature vector, and similarity is measured using the Manhattan distance.

3. Proposed Methodology

The proposed method introduces a multi-layer access control framework for cloud environments, incorporating improved DBSCAN clustering for efficient data retrieval and hybrid encryption (AES for data at rest and HE for data in use) to ensure

security.Data at rest refers to information stored in databases or storage systems, encrypted using AES for security. When needed, data is securely retrieved and processed at use through HE, enabling computations on encrypted data without decryption, preserving confidentiality throughout storage and operational phases.RMAAC model enforces fine-grained access permissions based on user roles, attributes, and data sensitivity. The primary challenges faced include ensuring efficient data retrieval while maintaining robust security, and overcoming the complexity of managing access control policies for diverse data types. These were addressed by optimizing the clustering algorithm, enhancing encryption methods, and implementing flexible access policies. Figure 1 depicts the overall proposed architecture.

3.1. Data Organization and Clustering

The input data, encompassing text, image, and video content, is first received and organized to streamline processing. The Improved DBSCAN algorithm is then applied to cluster the data efficiently. This enhanced clustering technique groups similar data points based on predefined similarity criteria, such as textual content, image features, or video characteristics. The algorithm's ability to handle noise and detect arbitrary-shaped clusters ensures robust data grouping, even with diverse media formats. By organizing data into meaningful clusters, this step facilitates faster and more accurate retrieval, significantly enhancing the system's ability to manage and process large, heterogeneous datasets in cloud environments.

DBSCAN

DBSCAN is a clustering algorithm commonly used in data mining and machine learning. Unlike traditional clustering algorithms like K-means, DBSCAN does not require the user to specify the number of clusters in advance. Instead, it identifies clusters based on the density of data points in the feature space.DBSCAN offers several advantages, including its ability to detect clusters of any shape and size without requiring prior knowledge of the number of clusters. It also excels at handling outliers and noisy data. However, it is sensitive to the selection of parameters like epsilon and minPts, struggles with clusters of varying densities, and can be computationally demanding for large datasets. DBSCAN identifies clusters by searching for points within a defined radius (ε) around a starting point. If the neighbors meet a minimum density ($MinPts$), they form a cluster. The process recursively expands clusters by adding reachable neighbors. Spatial

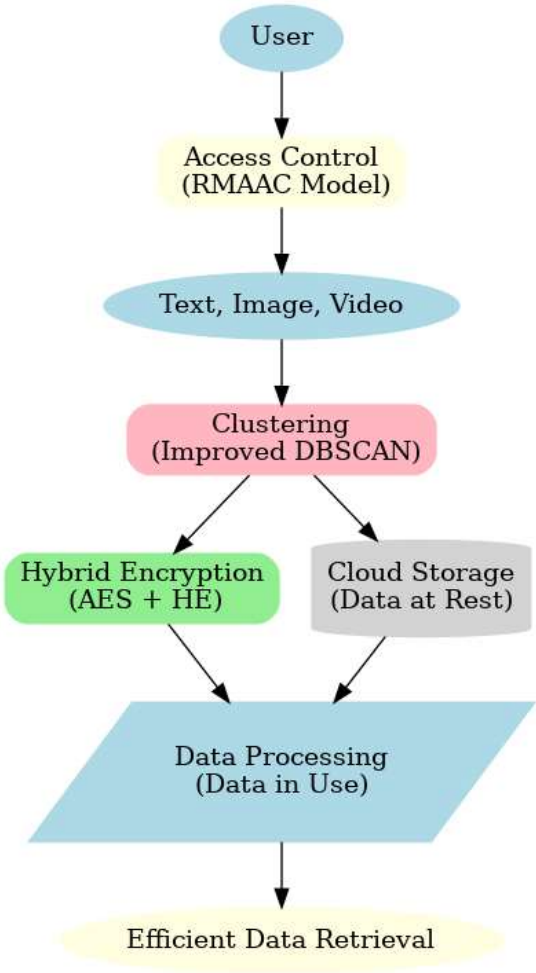


Figure 1. *Overall Proposed Architecture*

indexing optimizes the search, ensuring efficient handling of noise and irregular shapes.

Despite its limitations, DBSCAN remains a versatile choice for discovering clusters based on density, especially in scenarios with unknown cluster counts or irregular cluster shapes. This is a concise description of how DBSCAN functions:

Density-Based Clustering: DBSCAN operates under the assumption that clusters are areas of high density separated by areas of low density. It groups together data points that are closely packed, while marking points in sparse regions as outliers.

Key Parameters: Epsilon (ε) - This parameter defines the radius within which to search for neighbouring points around a given data point, and Minimum Points (minPts) - This parameter specifies the minimum number of points required within the epsilon radius to consider a data point as a core point.

Core Points: A data point is considered a core point if there are at least 'minPts' number of points (including itself) within its epsilon neighbourhood.

Border Points: Border points are not core points themselves but are within the epsilon radius of a core point. They typically belong to the same cluster as the core point.

Outliers: Data points that do not meet the criteria to be core or border points are considered outliers or noise. These are typically isolated points in low-density regions.

Algorithm Steps:

- Start with an arbitrary data point.
- Determine its epsilon neighborhood.
- If the neighborhood contains at least 'minPts' points, mark the point as a core point and expand the cluster by recursively adding all directly reachable points to the cluster.
- If the point does not have enough neighbors to be a core point but lies within the epsilon neighborhood of a core point, label it as a border point.

Iterate through all points in the dataset, expanding clusters and marking outliers as necessary.

3.2. Hybrid Encryption for Data Security

For data at rest, AES is applied to encrypt stored data, ensuring its confidentiality and protection against unauthorized access. For data in use, HE is employed, enabling computations on encrypted data without the need for decryption. This ensures security during processing while preserving data privacy. Together, these encryption techniques provide robust protection for sensitive data in cloud environments during both storage and active usage.

AES

AES is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. It is widely used today as it offers stronger security compared to its predecessor, the Data Encryption Standard (DES), and its variants like Triple DES (3DES). AES operates as a block cipher, encrypting data in fixed-size blocks. It accepts key sizes of 128, 192, or 256 bits. The encryption process involves several rounds, with each round consisting of four main steps:

- Sub Bytes: Each byte in the block is substituted with another byte using a predefined substitution table (S-box).
- Shift Rows: The rows of the block are shifted cyclically to the left.
- Mix Columns: Each column of the block is transformed using a matrix multiplication operation.
- Add Round Key: Each byte of the block is combined with a byte of the round key using bitwise XOR.

The number of rounds performed depends on the key size: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. AES encryption ensures data confidentiality and security, making it suitable for various applications such as wireless security, database encryption, secure communications, and file encryption. Its robustness against cryptographic attacks and its widespread adoption in both hardware and software implementations make it a cornerstone of modern cryptographic systems.

Homomorphic Encryption

HE is a revolutionary cryptographic method designed to enable computations on encrypted data without necessitating decryption, thereby ensuring data confidentiality even during processing by untrusted entities such as cloud service providers. Unlike traditional encryption systems, which require data to be decrypted before processingthus exposing sensitive information, HE allows operations to be performed directly on ciphertext. This unique capability ensures that the data remains secure throughout the computational process. The encrypted results, when decrypted, yield the same outcome as if the computations had been performed on the original plaintext data, offering a seamless blend of privacy and functionality.

Operational Mechanism

The HE framework operates through four critical stages. The first stage is Key Generation, where a cryptographic key pair is created. This pair includes a public key, which is used for encrypting data, and a private key, which is essential for decryption. The public key allows users to transform plaintext data into ciphertext during the Encryption stage. This ciphertext is a secure, unintelligible representation of the data that can be safely shared with third-party systems without risking privacy breaches.In the Evaluation phase, the system performs computations directly on the ciphertext. Mathematical operations, such as addition or multiplication, are executed on the encrypted data without ever revealing the underlying plaintext. The resulting ciphertext encapsulates the outcome of these computations. This capability allows third-party platforms, such as cloud servers, to process sensitive data securely without gaining access to the actual information. For instance, in healthcare or finance, computations on sensitive datasets can be conducted while ensuring regulatory compliance and protecting user privacy.

Finally, in the Decryption phase, the output ciphertext is decrypted using the private key. This step reveals the plaintext result of the computations, corresponding to the operations performed during the evaluation phase. The decrypted result is identical to what would have been obtained if the operations were directly applied to the plaintext data, thus ensuring both accuracy and confidentiality.HE's ability to preserve privacy during computations has

made it a cornerstone technology in domains requiring secure data processing. By eliminating the need to expose data during processing, HE offers a robust framework for privacy-preserving computation, paving the way for secure, efficient, and trustworthy data-driven services.

The equations for encryption and decryption in the HE scheme is given as per Eq. (1) and Eq.(2):

$$c = g^m \cdot r^a \, mod \, a^2 \quad (1)$$

$$m = \left(\frac{c^\lambda mod \, a^2 - 1}{a} \cdot \mu\right) mod \, a \quad (2)$$

Where, c represents cipher text, m is message in plain text, r represents random number, (a, g) represents public key, (λ, μ) defines private key.

3.3. Access Control Framework

RMAAC framework is designed to enhance traditional access control mechanisms by incorporating both user roles and specific attributes, along with the sensitivity level of the data being accessed. The main objective of this model is to ensure that access permissions are not only determined based on a user's role but also take into account additional attributes, such as environmental conditions or security levels, thus providing a more granular and dynamic approach to data access. In the proposed framework, the access policy is defined as a RMAAC model. It grants permissions based on users' roles, specific attributes, and the sensitivity of the data being accessed. This fine-grained approach ensures secure, flexible access, prevents unauthorized actions, and adapts policies to varying data types, enhancing security and usability.

Key Components of RMAAC Model

The RMAAC model consists of four main components: Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Information Point (PIP), and Policy Administration Point (PAP). These components work together to enforce access policies that regulate how data is accessed and shared within a system.

- Policy Enforcement Point (PEP): This component is responsible for intercepting an access request from an authenticated subject (user or entity) and forwarding it to the PDP for evaluation. Once the access request is evaluated, the PEP enforces the decision by either granting or denying access.
- Policy Decision Point (PDP): The PDP is tasked with evaluating the access request by checking the corresponding policies, which are stored and managed by the PAP. It uses attributes (obtained via the PIP) to assess whether the request should be granted, based on defined Digital Policies (DPs) and Meta Policies (MPs).
- Policy Information Point (PIP): The PIP provides the necessary attribute values (e.g., security levels, environmental conditions) required for access policy evaluation. It supplies the PDP with real-time or static attribute information that determines whether the conditions for access are met.
- Policy Administration Point (PAP): The PAP is responsible for creating, managing, and updating policies, including the DPs and MPs. These policies guide the PDP's decision-making process in evaluating access requests.

While the original RMAAC model offers robust access control, it lacks a mechanism to ensure data confidentiality. To address this, the proposed RMAAC model introduces additional measures to secure data during access and computation. The cloud server (CS) in the model determines the security level of the data user before searching for and retrieving data from the database, thus preventing inefficient and lengthy database searches. This improves the system's performance and scalability while maintaining data confidentiality.

System Architecture

The proposed RMAAC architecture, involves five main entities:

- Attribute Authority Server (AAS): A trusted entity that generates system parameters and secret keys for data users. The AAS also manages security levels and ensures proper access control.
- Identity and Access Management Server (IAMS): The IAMS generates tokens corresponding to users' security levels based on predefined sets. It works in conjunction with the AAS to create, update, and manage security levels.
- Cloud Server (CS): The CS stores encrypted data (ciphertexts) and acts as an intermediary to check the validity of access tokens. It ensures that only authorized users can access the data based on predefined policies.
- Data Owner (DO): The DO is an IoT data producer who defines the security level of the data and generates encrypted ciphertexts, along with associated metadata. The DO uploads this encrypted data to the CS for sharing.
- Data User (DU): The DU is an IoT device requesting access to the encrypted data. It presents a token to the CS for verification and, if the DU satisfies the policy, decrypts the data to obtain the original message.

In the system, the PAP, PIP, PEP, and PDP components work together within the architecture. The PAP relies on both the AAS and the DO for policy management, while the PIP involves the DO,

CS, and DU for attribute management. The PEP and PDP components, which handle the enforcement and decision-making processes, are represented by IAMS, CS, and DO, ensuring the consistency of the RMAAC model. This integrated system provides a secure and efficient method for managing access control in IoT environments, ensuring both data confidentiality and effective policy enforcement.

3.4. Data Retrieval and Filtering

Efficient data retrieval is crucial in systems that handle large volumes of information, especially when real-time access is needed. One approach to achieve this is by leveraging clustering techniques, which group similar data items together based on predefined characteristics. These clusters can then be used to filter and expedite the retrieval process. In this approach, clustering divides the data into meaningful groups or clusters. Once these clusters are created, the system can quickly identify which cluster contains the relevant data for a given query. Clustering filtering techniques help by narrowing the search to only those clusters that are most likely to contain the requested content, thus significantly reducing the search space. When a user requests data, the system first determines which cluster the query matches. By utilizing predefined cluster properties or metadata, it can avoid searching the entire dataset, focusing only on the relevant cluster. This results in faster response times, as the retrieval process bypasses irrelevant data.

Authorization and Access Granting

Upon receiving a data retrieval request, the access control mechanism evaluates several factors to determine whether to grant or deny access. First, it checks the user's role, which defines the actions they are permitted to perform within the system based on their assigned responsibilities or job functions. For example, an administrator may have broader access rights compared to a regular user. In addition to the role, the system also considers the user's attributes, such as their identity, credentials, or other personal characteristics. These attributes provide further context for the request, helping refine the decision-making process; for instance, a user's security clearance, location, or the time of access may influence whether they can retrieve certain data. Finally, the sensitivity level of the data being requested is assessed. Data is often classified into categories like public, confidential, or restricted, and more sensitive data requires stricter access controls. The access control mechanism then compares the sensitivity of the data with the user's role, attributes, and security clearance, ensuring that only authorized individuals with the appropriate context are allowed to access sensitive or restricted information. This multi-layered approach ensures that data access is granted securely and appropriately, based on a combination of user-specific and data-specific factors.

Data Use and Secure Processing

To process and manipulate data securely, HE is employed, allowing operations to be performed directly on encrypted data without the need for decryption. This ensures that sensitive information remains confidential even during computation, preserving privacy throughout the data processing lifecycle. The primary advantage of HE is its ability to perform arithmetic or logical operationssuch as addition, multiplication, or more complex algorithmson ciphertext, which represents encrypted data. These operations produce encrypted results, and when decrypted, the output is the same as if the operations had been performed on the original, unencrypted data.

For example, in a healthcare system, HE can be used to analyze encrypted medical data, such as patient records, without exposing the actual data. The data is first encrypted using a public key and sent to a server for processing. The server performs computations on the encrypted data and returns the encrypted results. Once the results are returned, the data owner, with the private key, decrypts the output to obtain the final, meaningful result.By maintaining data in its encrypted form throughout the process, HE ensures data privacy and security, mitigating risks related to unauthorized access or data breaches during computation. This makes it particularly valuable in cloud computing environments or any situation where sensitive data must be processed without exposing it to third parties.

The RMAAC model ensures that data is provided only to authorized users by evaluating their roles, attributes, and the data's sensitivity. When a request is made, the system checks the user's credentials and the data's access policies. The Policy Decision Point (PDP) assesses these factors, while the Policy Enforcement Point (PEP) grants or denies access based on predefined rules. This method ensures data is securely accessed only by users with appropriate permissions, maintaining confidentiality and integrity. Continuous monitoring of the system involves assessing the retrieval speed, security, and clustering performance to ensure optimal functionality. The system tracks the time taken for data retrieval, evaluates the effectiveness of security protocols, and measures the accuracy of clustering algorithms. Regular evaluations help identify performance bottlenecks or security vulnerabilities. Based on these insights, necessary adjustments, such as optimizing retrieval algorithms or enhancing security measures, are made to maintain system

efficiency, ensuring reliable and secure data access for users.

4. Result and Discussion

The evaluation of the proposed Multi-Layer Access Control (MLAC) system for cloud data demonstrated its effectiveness in securing diverse data types, including text, images, and videos. The system's performance was assessed based on encryption time, decryption time, and the dataset sizes for text, image, and video data. The datasets used for experimentation were fetch_20newsgroups for text, CIFAR-10 for image, and CIFAR-10 video for video data. The results are analyzed to provide insights into the system's efficiency and scalability.

Encryption and Decryption Time

The encryption and decryption times varied depending on the data type and size, reflecting the computational complexity associated with each format. For text data from the fetch_20newsgroups dataset, the encryption and decryption times were observed to be the fastest, given the smaller dataset size in kilobytes and the relatively low processing requirements of text data. For image data, CIFAR-10 introduced moderate complexity due to the higher data density and multi-dimensional structure of the images. Encryption and decryption times for video data were the highest, as CIFAR-10 video datasets inherently require processing of sequential frames with substantial data volume. This trend aligns with the expectations that encryption and decryption complexity increase with data size and format intricacy.

Dataset Size

The dataset sizes played a significant role in influencing the observed encryption and decryption times. Text datasets from fetch_20newsgroups were compact, typically measured in kilobytes, which allowed for rapid processing. CIFAR-10 images, though larger than text, remained within a manageable range, ensuring reasonable processing times. However, CIFAR-10 video data represented the most demanding workload, as the dataset size was significantly larger, given its sequential nature and storage requirements for multiple frames.

Analysis of Results

The results highlight the scalability of the MLAC system across varying data types and sizes. The efficient processing of smaller datasets like text underscores the system's suitability for lightweight applications, while the effective handling of larger datasets like video demonstrates its robustness. The balanced performance across different data types illustrates the adaptability of the MLAC system, making it viable for diverse cloud-based applications. Overall, the proposed MLAC system achieved efficient encryption and decryption for text, image, and video data, ensuring data security without compromising processing speed. These results affirm the system's capability to meet the security and performance needs of modern cloud environments, offering a reliable solution for multi-layer access control in diverse use cases.

4.1. Dataset Collection

The CIFAR-10 dataset [26] is a popular collection for image classification tasks, consisting of 60,000 32x32 pixel colour images divided into 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. It includes 50,000 training images and 10,000 testing images. This dataset is widely used for benchmarking image classification models and evaluating their accuracy.The fetch_20newsgroups dataset [27] is commonly used for text classification tasks. It contains 18,846 documents from 20 different newsgroups. These documents are divided into training and test subsets, with options for removing metadata such as headers, footers, or quotes to improve model performance. Each document is categorized into one of 20 classes, making it a valuable resource for evaluating classification algorithms.

4.2. Performance Measures

The performance measures (Encryption Time, and Decryption Time) are essential for evaluating the efficiency and effectiveness of cryptographic algorithms in securing data in cloud environments.

Encryption Time

Encryption time refers to the amount of time required to encrypt a given dataset. This is a critical performance metric as it directly impacts the speed of data storage and retrieval in cloud systems. A lower encryption time indicates better system performance. The formula for encryption time is given as per Eq. (3).

$$ET = \frac{total\ time\ taken\ to\ encrypt}{datasize} \quad (3)$$

Where the total time taken to encrypt is the time taken by the encryption algorithm to encrypt the dataset, and datasize refers to the volume of data being encrypted (usually in bytes or kilobytes).

Decryption Time

Decryption time is the amount of time required to decrypt the data back to its original form after encryption. This metric is crucial for evaluating the speed and usability of encrypted data in a system. Just like encryption time, a lower decryption time is preferred for faster data retrieval. The formula for encryption time is given as per Eq. (4).

$$DT = \frac{total\ time\ taken\ to\ decrypt}{datasize} \quad (3)$$

Where total time taken to decrypt is the time taken by the decryption algorithm to decrypt the data, and datasize refers to the volume of data being decrypted.

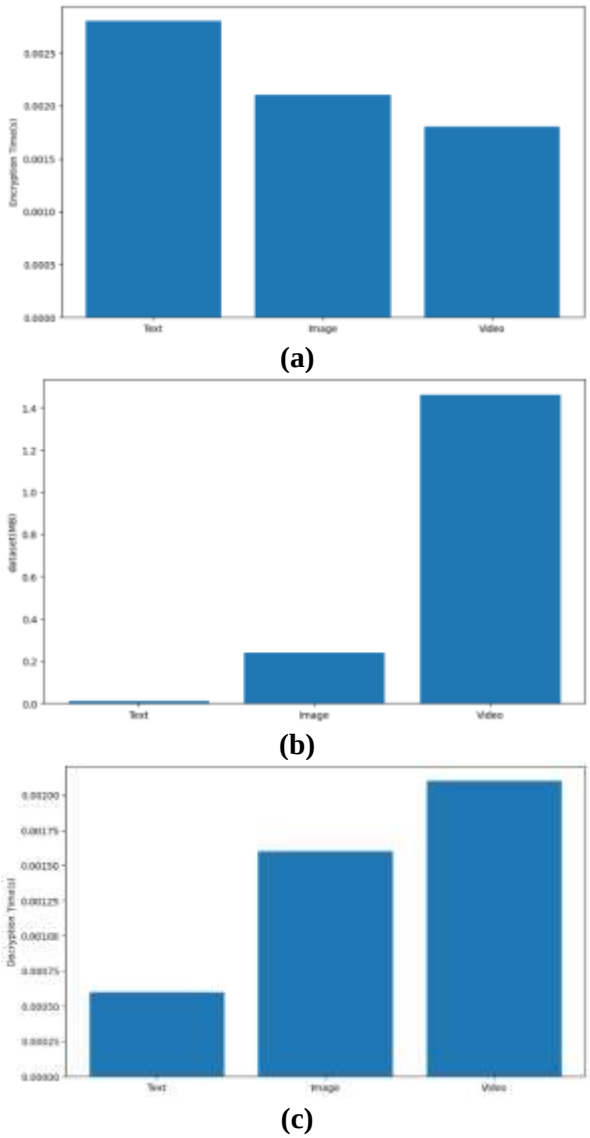
Execution Time

Execution Time refers to the total time taken for a system or model to complete a particular process, such as encryption or decryption. In the context of encryption algorithms, it includes the time spent on all operations from the beginning to the end of the process, including setup, key generation, encryption/decryption, and any additional tasks. The formula for execution time is given as per Eq. (5).

$$execution\ time = t_{setup} + t_{enc} + t_{dec} + t_{oo} \quad (5)$$

Where, $t_{setup}$is the time taken for initializing the encryption system, such as key generation or algorithm setup, $t_{enc}$is the time required to encrypt the data, $t_{dec}$is the time required to decrypt the data, andt_{oo}refers to any additional tasks or processes, such as key management or validation, that contribute to the overall execution time.

The results show the storage and processing efficiency of different data types in terms of file sizes. The image, with a size of 20 MB, is relatively smaller and likely requires less bandwidth and quicker processing time. The video, at 100 MB, is significantly larger, which could lead to longer loading times, higher network utilization, and increased storage requirements. Text, sized at 40 MB, falls in between, balancing storage needs and processing time. These varying file sizes highlight the need for optimization in handling multimedia data.



(a)

(b)

(c)

Figure 2. *Graphical Representation (a) Encryption Time, (b) Dataset Size (c) Decryption Time*

(a)

(b)

Figure 3. *Comparison of Image and Video Data*

Figure 3 compares image and video data, highlighting differences in file size and processing demands. The image, at 20 MB, requires less bandwidth and storage, while the video, at 100 MB, demands more.

Table 1. *Existing vs Proposed Model*

Metrics	TDES [17]	CP-ABE [14]	IRS [23]	Proposed
Execution Time (minutes)	48	30	50	20
Encryption (seconds)	20	30	50	10
Decryption (seconds)	20	40	60	10

Table 1 compares the performance of the existing encryption models (TDES, CP-ABE, IRS) with a Proposed Model based on three key metrics: Execution Time, Encryption Time, and Decryption Time. The Proposed Model shows significant improvements in all areas. It has the shortest Execution Time (20 seconds), outperforming TDES (48 seconds), CP-ABE (30 seconds), and IRS (50 seconds). Similarly, its Encryption Time (10 seconds) is the fastest, compared to TDES (20

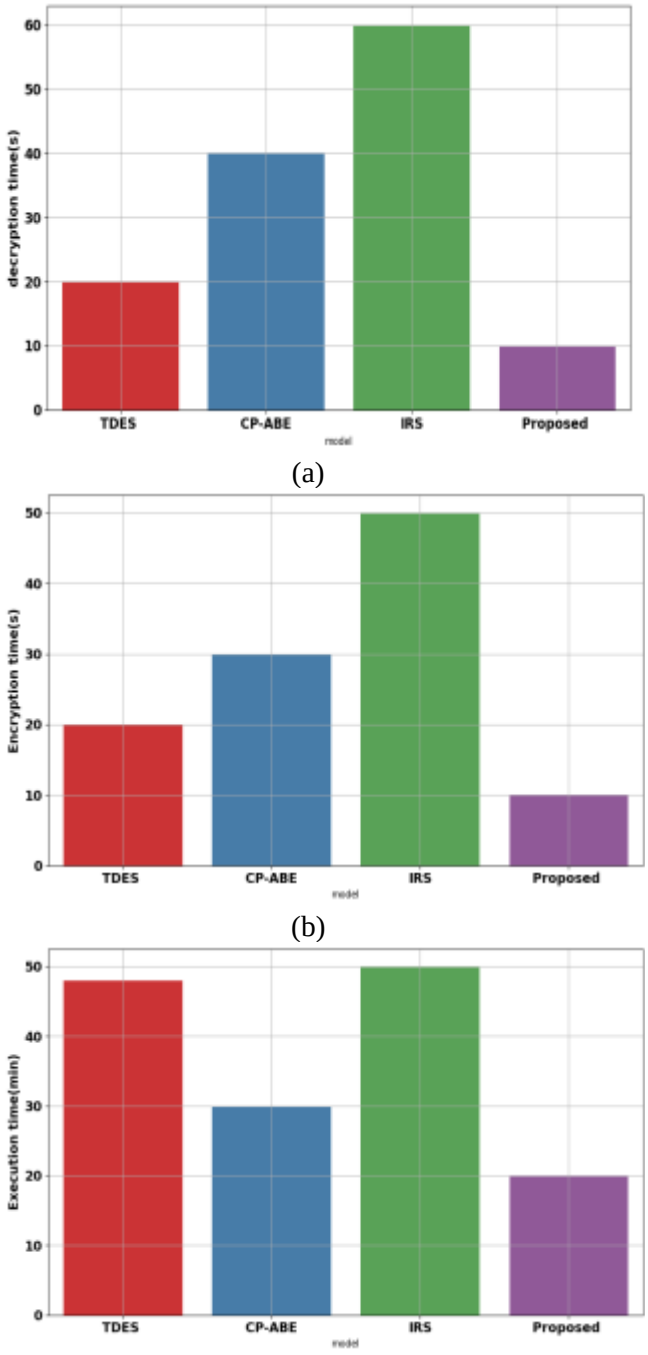


Figure 4. *Graphical Representation of Existing vs Proposed Model*

seconds), CP-ABE (30 seconds), and IRS (50 seconds). The Decryption Time is also minimal at 10 seconds, far quicker than TDES (20 seconds), CP-ABE (40 seconds), and IRS (60 seconds). Overall, the Proposed Model demonstrates superior efficiency and performance.

Figure 4 illustrates a comparison between the existing models (TDES, CP-ABE, IRS) and the Proposed Model, showcasing their performance differences in Execution Time, Encryption Time, and Decryption Time through a graphical representation.

5. Conclusion

Using an Improved DBSCAN algorithm and improved clustering filtering techniques, this research developed a unique multi-layer access control system for cloud environments that improves search performance across several data layers. This clustering technique effectively arranged data according to similarity, allowing for the rapid and precise retrieval of text, picture, and video material. By integrating AES for data at rest and HE for data in use, the framework ensured data security and privacy while permitting safe data manipulation without threatening secrecy.By implementing RMAAC model, which issued permissions according to a user's role, attributes, and the degree of sensitivity of the data being accessed, the access control mechanism was further reinforced. This fine-grained control allowed for configurable policies for various data kinds while limiting unwanted access. According to simulation findings, the suggested architecture greatly enhanced clustering performance, security, and data retrieval speed, making it a useful option for cloud storage systems that manage a variety of media types.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** This work has been financially supported by the RUSA – Phase 2.0 grant, as sanctioned in Letter No. F. 24-51/2014-U, Policy (TNMulti-Gen), Department of Education, Government of India, dated October 9, 2018.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Mostafa, A.M., Ezz, M., Elbashir, M.K., Alruily, M., Hamouda, E., Alsarhani, M. and Said, W., (2023). Strengthening cloud security: an innovative multi-

factor multi-layer authentication framework for cloud user authentication. *Applied Sciences*, 13(19),10871. https://doi.org/10.3390/app131910871

[2] Awadh, W.A., Alasady, A.S. and Hashim, M.S., (2023). A multilayer model to enhance data security in cloud computing. Indonesian *Journal of Electrical Engineering and Computer Science*, 32(2),1105-1114.

[3] Gupta, M., Bhatt, S., Alshehri, A.H. and Sandhu, R., (2022). Access control models and architectures for IoT and cyber physical systems (pp. 1-173). *Cham, Switzerland: Springer*.

[4] Adee, R. and Mouratidis, H., (2022). A dynamic four-step data security model for data in cloud computing based on cryptography and steganography. *Sensors*, 22(3),1109. https://doi.org/10.3390/s22031109

[5] Alemami, Y., Al-Ghonmein, A.M., Al-Moghrabi, K.G. and Mohamed, M.A., (2023). Cloud data security and various cryptographic algorithms. *International Journal of Electrical and Computer Engineering*, 13(2),1867. http://doi.org/10.11591/ijece.v13i2.pp1867-1879

[6] Chaudhry, S.A., Yahya, K., Al-Turjman, F. and Yang, M.H., (2020). A secure and reliable device access control scheme for IoT based sensor cloud systems. *IEEE Access*, 8,139244-139254. doi: 10.1109/ACCESS.2020.3012121

[7] Yang, C., Tan, L., Shi, N., Xu, B., Cao, Y. and Yu, K., (2020). AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud. *IEEE Access*, 8,70604-70615. doi: 10.1109/ACCESS.2020.2985762

[8] Qin, X., Huang, Y., Yang, Z. and Li, X., (2021). A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing. *Journal of Systems Architecture*, 112,101854. https://doi.org/10.1016/j.sysarc.2020.101854

[9] Qi, S., Lu, Y., Wei, W. and Chen, X., (2020). Efficient data access control with fine-grained data protection in cloud-assisted IIoT. *IEEE Internet of Things Journal*, 8(4), pp.2886-2899.

[10] Egala, B.S., Pradhan, A.K., Badarla, V. and Mohanty, S.P., (2021). Fortified-chain: a blockchain-based framework for security and privacy-assured internet of medical things with effective access control. *IEEE Internet of Things Journal*, 8(14),11717-11731.

[11] Han, D., Zhu, Y., Li, D., Liang, W., Souri, A. and Li, K.C., (2021). A blockchain-based auditable access control system for private data in service-centric IoT environments. *IEEE Transactions on Industrial Informatics*, 18(5),3530-3540.

[12] Mangalagowri, R. and Venkataraman, R., (2023). Ensure secured data transmission during virtual machine migration over cloud computing environment. *International Journal of System Assurance Engineering and Management*, pp.1-12. DOI:10.1007/s13198-022-01834-8

[13] Susilabai, S.S., Mahendran, D.S. and Peter, S.J., (2022). A trusted user integrity-based privilege access control (UIPAC) for secured clouds. In *Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021* (pp. 499-520). Springer Singapore.

[14] Fugkeaw, S., (2020). A fine-grained and lightweight data access control model for mobile cloud computing. *IEEE Access*, 9,836-848. doi: 10.1109/ACCESS.2020.3046869

[15] Xu, G., Xu, S., Ma, J., Ning, J. and Huang, X., (2023). An Adaptively Secure and Efficient Data Sharing System for Dynamic User Groups in Cloud. *IEEE Transactions on Information Forensics and Security*. 18, 5171-5185. DOI:10.1109/TIFS.2023.3305870

[16] Anju, J. and Shreelekshmi, R., (2022). A faster secure content-based image retrieval using clustering for cloud. *Expert Systems with Applications*, 189,116070. https://doi.org/10.1016/j.eswa.2021.116070

[17] Ramachandra, M.N., Srinivasa Rao, M., Lai, W.C., Parameshachari, B.D., Ananda Babu, J. and Hemalatha, K.L., (2022). An efficient and secure big data storage in cloud environment by using triple data encryption standard. *Big Data and Cognitive Computing*, 6(4),101. https://doi.org/10.3390/bdcc6040101

[18] Rafique, A., Van Landuyt, D., Beni, E.H., Lagaisse, B. and Joosen, W., (2021). CryptDICE: Distributed data protection system for secure cloud data storage and computation. *Information Systems*, 96,101671. https://doi.org/10.1016/j.is.2020.101671

[19] Huang, Z., Zhang, M. and Zhang, Y., (2019). Toward efficient encrypted image retrieval in cloud environment. *IEEE Access*, 7,174541-174550. doi: 10.1109/ACCESS.2019.2957497

[20] Xu, Y., Zhao, X. and Gong, J., (2019). A large-scale secure image retrieval method in cloud environment. *IEEE Access*, 7,160082-160090. doi: 10.1109/ACCESS.2019.2951175

[21] Li, J.S., Liu, I.H., Tsai, C.J., Su, Z.Y., Li, C.F. and Liu, C.G., (2020). Secure content-based image retrieval in the cloud with key confidentiality. *IEEE Access*, 8, 114940-114952.

[22] Chai, X., Wang, Y., Gan, Z., Chen, X. and Zhang, Y., (2022). Preserving privacy while revealing thumbnail for content-based encrypted image retrieval in the cloud. *Information Sciences*, 604,115-141. https://doi.org/10.1016/j.ins.2022.05.008

[23] Shen, M., Cheng, G., Zhu, L., Du, X. and Hu, J., (2020). Content-based multi-source encrypted image retrieval in clouds with privacy preservation. *Future Generation Computer Systems*, 109,621-632.

[24] Xu, Y., Gong, J., Xiong, L., Xu, Z., Wang, J. and Shi, Y.Q., (2017). A privacy-preserving content-based image retrieval method in cloud environment. *Journal of Visual Communication and Image Representation*, 43,164-172.

[25] Xia, Z., Wang, L., Tang, J., Xiong, N.N. and Weng, J., (2020). A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing. *IEEE Transactions on Network Science and Engineering*, 8(1),318-330. DOI:10.1109/TNSE.2020.3038218

[26] Dataset taken for image and video from: "https://www.kaggle.com/competitions/mu-cifar10", dated 1/11/2024.
[27] Dataset taken for text from: "https://scikit-learn.org/dev/modules/generated/sklearn.datasets.fetch_20newsgroups.html", dated 1/11/2024.