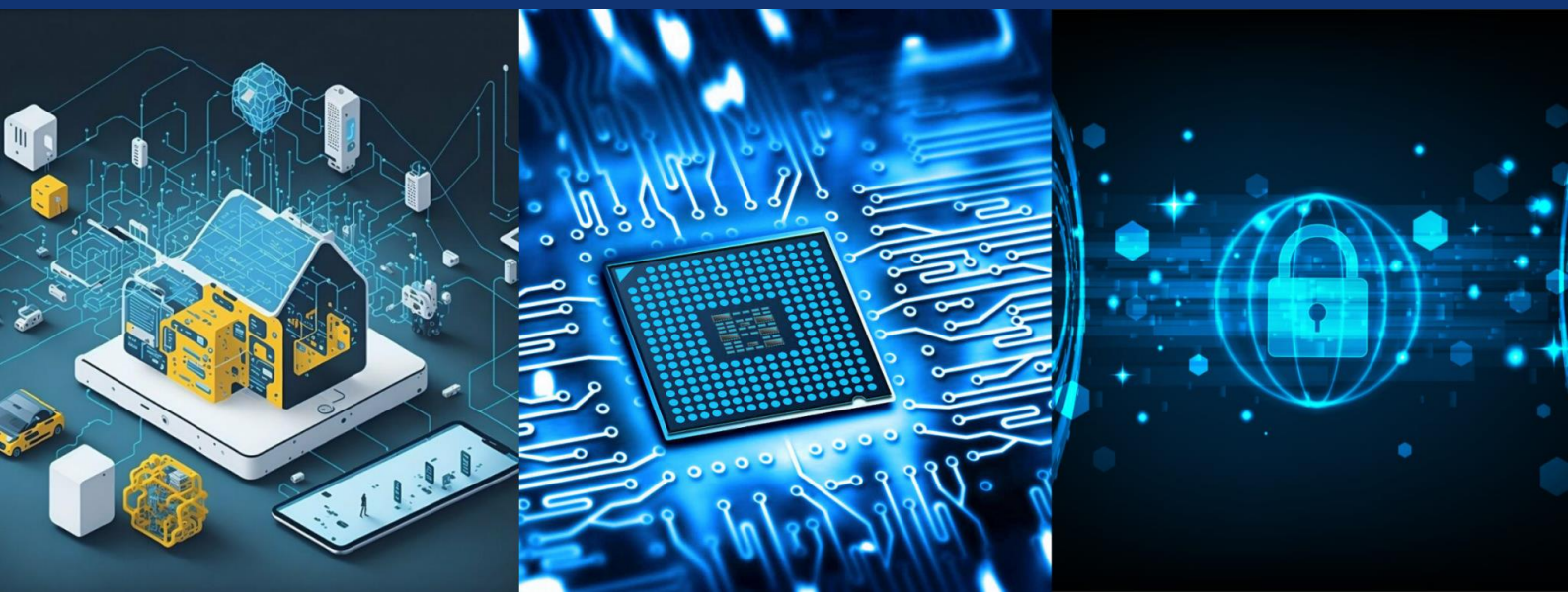
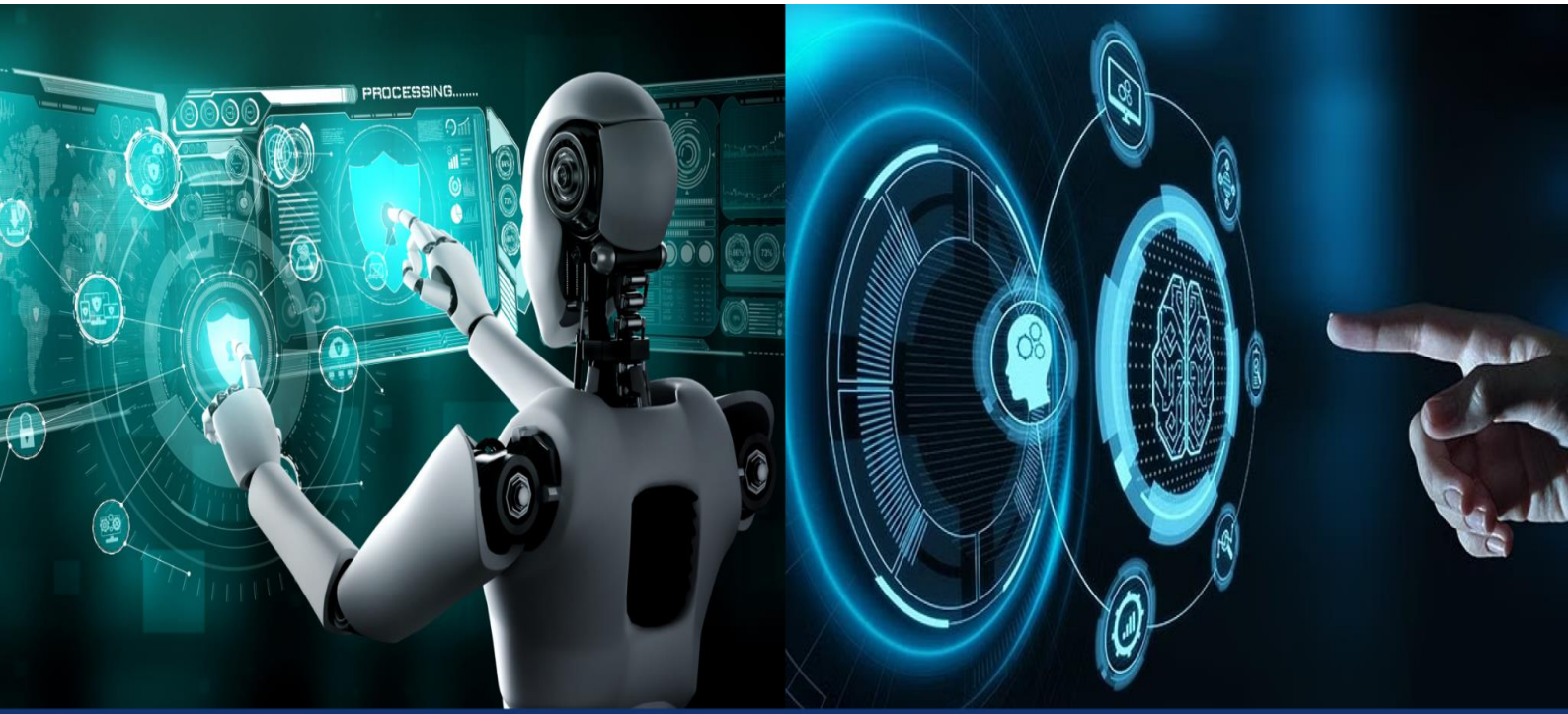
ISSN(O): 2320-9801
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771 **Volume 13, Issue 4, April 2025**

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304128

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Design and Implementation of end-to-end Secure Data Management in Multi-Authority Cloud Environments

Swapnali S. Bhokare [1], Prof. Sachin B. Bhosale [2], Prof. Anand A. Khatri [3]

Department of Computer Engineering, Jai Hind College of Engineering, Maharashtra, Kuran, Pune, India[1,2,3]

ABSTRACT: This proposed system explores key aspects of cloud computing and cloud file services, with a strong focus on usability and storage optimization. At its core, it looks into how data de-duplication can improve the experience for both cloud service providers and users. The system proposes an efficient method for identifying and removing duplicate files using file checksum algorithms, which offer faster performance than traditional techniques. To maintain security and access control, the system ensures that users are granted specific privileges through unique access tokens. Built on a hybrid cloud model, the solution is designed to reduce resource usage while improving reliability. Compared to conventional de-duplication methods, this approach significantly cuts down on system overhead during duplicate data removal. The project also dives into both content-level and file-level de-duplication strategies to make the most of available cloud storage, aiming for a smarter, more efficient way to manage data in the cloud.

KEYWORDS: Cloud storage, Data De-duplication, Checksum Algorithms, Hybrid Cloud, Security, File-Level Deduplication

I. INTRODUCTION

With the explosion of digital data in recent years—around 1 to 2 terabytes being generated in just 2019 and 2020 alone—managing storage has become a serious challenge. One of the biggest issues facing large-scale storage systems today is how to make storage more efficient and cost-effective. In situations where bandwidth is limited, data de-duplication has emerged as a smart solution. By identifying and eliminating duplicate data, this technique not only cuts down on unnecessary data transmission but also frees up valuable storage space.

Cloud computing, which continues to evolve rapidly in the tech world, has embraced data de-duplication as a core method to improve storage efficiency. As digital infrastructure grows more complex, so does the way organizations use the cloud.

Enter multi-cloud strategies—a trend that's gaining real momentum. More and more companies are choosing to use multiple cloud service providers rather than sticking with just one. This approach lets them tap into the unique benefits of each platform, offering more flexibility, better scalability, and reducing their dependence on any single vendor. It also helps cushion the blow of unexpected service disruptions[1]. As businesses grow increasingly dependent on cloud services to run their critical operations, getting a handle on how to manage databases across multiple cloud environments has never been more important.[2].

1.1 Definition and Characteristics of Multi-Cloud Databases

Multi-cloud databases are systems that operate across multiple cloud platforms. This setup allows organizations to spread their data and applications among different cloud service providers, giving them more flexibility and control. Some key features of multi-cloud databases include:

- **Data Distribution:** Data can be spread across several cloud platforms, which helps improve performance and reduce delays [1].
- **Scalability:** These databases can be easily scaled up or down depending on workload demands, ensuring smooth performance even during traffic spikes [1].
- **Redundancy and Resilience:** Using multiple cloud providers builds in redundancy. This means if one provider experiences an outage, the others can keep things running, minimizing the risk of data loss [3].

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304128

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

1.2 Why Organizations are Embracing Multi-Cloud

There are several compelling reasons why companies are moving toward a multi-cloud strategy for managing their databases:

- **Cost Savings:** By comparing prices across providers, organizations can select the most cost-effective options and reduce overall cloud expenses [4].
- **Freedom from Vendor Lock-in:** Relying on just one cloud provider can be risky. Multi-cloud strategies reduce dependency on any single vendor, giving organizations more flexibility and leverage [5].
- **Regulatory Compliance:** Some regulations require that data be stored in specific regions. A multi-cloud approach makes it easier to comply with these rules by choosing providers with global infrastructure [6].

1.3 Security Challenges in Multi-Cloud Database Management

While the benefits of multi-cloud environments are significant, they also introduce complex security challenges that need careful attention. The interconnected nature of multiple clouds, each with its own standards and policies, creates new risks [7].

- **Bigger Attack Surface:** Using multiple cloud services means more points of access—and potentially more vulnerabilities. Every new integration could open the door to a security threat, so it's vital for organizations to have a thorough and unified security plan [3].
- **Data Fragmentation and Consistency Issues:** When data is spread out across different clouds, keeping it synchronized and consistent can be tough. It's essential to have solid governance and synchronization strategies in place to ensure accuracy and integrity [8].
- **Inconsistent Security Policies:** Each cloud provider may have different security standards, which can make it hard to apply uniform protection across the board. Organizations need a consistent, cross-platform security framework to close these gaps [9].
- **The Shared Responsibility Model:** Security in the cloud is a joint effort. While cloud providers secure the infrastructure, the responsibility for data protection lies with the organization. In a multi-cloud setup, this model becomes more complex and requires careful coordination [10].
- **Limited Centralized Visibility and Control:** Managing security across different platforms can make it harder to get a clear, unified view of potential threats. Without centralized tools—like advanced SIEM solutions—it becomes difficult to monitor, detect, and respond to security incidents promptly [11].

This paper proposes an efficient way to perform de-duplication using file checksum extraction techniques, significantly reducing the time required to detect duplicate data. When a user uploads a file, the system computes its checksum and compares it to the existing checksums in the database. If a match is found, the system updates the entry; otherwise, it creates a new entry. The system encompasses cloud servers, data consumers, and owners, and ensures efficient data management by leveraging advanced de-duplication techniques.

II. HISTORY & BACKGROUND

Kaiping Xue et al. [12] introduced an innovative heterogeneous architecture aimed at addressing single-point performance bottlenecks while improving access control. A standout feature of this system is its built-in auditing mechanism, which enhances overall security. To ease the burden of verifying user legitimacy, the system leverages multiple attribute authorities. Once a user's legitimacy is confirmed, a central authority (CA) generates hidden keys for them. Unlike traditional multi-authority access control methods, this system processes each authority's attribute set independently. The added auditing layer is especially valuable, as it helps detect any malicious or improper behavior from the Attribute Authorities (AAs), bolstering system integrity.

Kan Yang et al. [13] proposed a revocable Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme within a multi-authority environment. Their solution supports both forward and backward security using an attribute revocation tool. In settings where different authorities issue attributes independently, their approach offers a robust, flexible, and secure framework for data access control.

Zhongma Zhu et al. [14] developed a secure anti-collusion key distribution technique that removes the need for third-party networks. Users can securely receive their private keys directly from the group owner. This method enables fine-grained access control and ensures revoked users cannot regain entry—even in cases where they may attempt to collude

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

with an untrusted cloud provider. The system uses polynomial-based revocation to manage access, eliminating the need for revoked users to revalidate their status in the future.

N. Attarpadung et al. [15] put forward a Key-Policy Attribute-Based Encryption (KP-ABE) system that supports non-monotonic access structures and maintains a constant ciphertext size. This was the first framework to introduce KP-ABE with negated attributes (non-monotonicity). Their model also introduces an identity-based revocation mechanism that, when combined with a monotonic construction, results in a powerful and expressive KP-ABE solution.

F. Zhang and K. Kim [16] focused their research on bilinear pairings and applied the Java Pairing Library to develop an ID-based ring signature system for secure access control. They evaluated both security and performance of their method in comparison to existing models, especially regarding encryption and decryption efficiency. The authors also presented various user access management policies aimed at preserving data owner privacy and confidentiality.

J. Han et al. [17] introduced the first identity-based threshold ring signature technique that doesn't depend on Java pairings. Their system ensures the privacy of individual signers even when the Public Key Generator (PKG) is in use. This flexible framework supports multiple levels of signer privacy, making it adaptable to real-world use cases.

J. Yu et al. [18] emphasized the importance of validating a system's security architecture before integrating it into a broader security framework. Their solution employs AES-128 encryption with a 16-bit key for end-to-end user authentication and secure data transmission.

Kan Yang [19] tackled the shortcomings of existing CP-ABE schemes in multi-authority cloud storage settings—particularly issues around inefficient decryption and limited revocation capabilities. His proposed solution, DAC-MACS (Data Access Management for Multi-Authority Cloud Storage), introduces a more efficient decryption model and a stronger attribute revocation mechanism that ensures forward and backward security.

Guangyan Zhang [20] proposed "CaCo," a Cauchy coding technique designed for secure and efficient data storage in the cloud. By leveraging Cauchy matrix heuristics and optimized XOR schedules, CaCo generates ideal coding schemes tailored for different redundancy levels. This approach, implemented within a distributed file system, outperforms systems like "Cloud 2.5" in terms of storage efficiency and data protection.

Ibrahim Adel [21] offered a new replica placement strategy for the Hadoop Distributed File System (HDFS). His method focuses on solving load balancing problems by evenly distributing data replicas across all nodes—without the need for external load balancing tools. Simulations showed that this strategy adheres to all HDFS replication rules and can significantly improve cluster efficiency, especially in uniformly provisioned environments.

III. DESIGN ISSUES

The proposed framework is designed to deliver efficient de-duplication while ensuring system stability. It supports both file-level and block-level de-duplication to enhance secure data management. When a user uploads a file, the system first checks for duplication at the file level. If the file already exists in the system, the storage server rejects it, saving space equal to the size of that file. If the file is unique, it's broken down into fixed-size blocks for further processing. These blocks are then fragmented and distributed across multiple storage nodes using secure Role-Based Access Control (RBAC) and secret sharing techniques. Before these blocks are uploaded, a second layer of de-duplication—block-level de-duplication—is performed. Any block that matches an existing one is not uploaded, thereby saving storage equivalent to the size of that block. Security is addressed from two critical angles:

- **Authorization for duplicate checking**
- **Confidentiality of stored data**

To achieve these goals, the system incorporates convergent encryption, symmetric encryption, and a Proof of Work (PoW) mechanism. These technologies help establish a secure and stable de-duplication process, ensuring that data remains protected throughout the transmission and storage phases.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304128

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[1] ***Architecture***

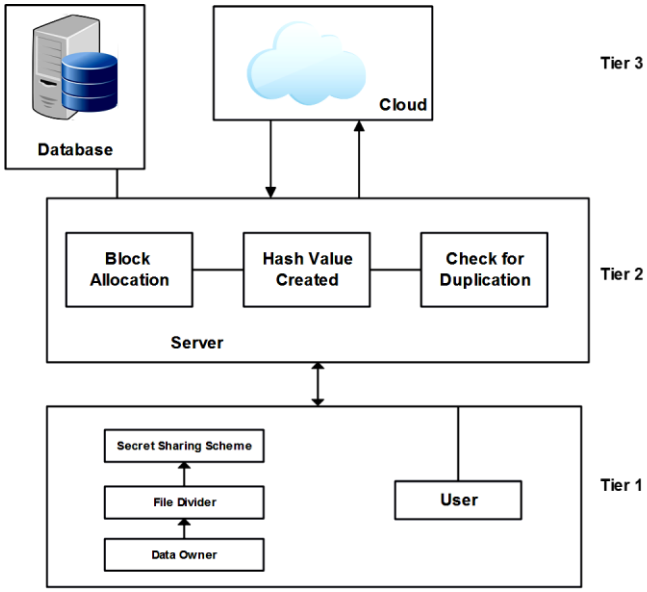


Figure 1. System Architecture

[2] ***Algorithms***

Algorithm 1: Hash Generation

The process of hash generation involves using a cryptographic hash function to create a unique hash value. Hashing is essential for ensuring data integrity and enabling de-duplication. The following algorithm implements a robust hashing technique to detect duplicate data.

Input: plain_data d,

Output: Generated hash value H_Values for the given data.

Step 1: Receive the plain data as a list d_list.

Step 2: Apply SHA-256 hashing function from the SHA family.

Step 3: Compute the hash value: CurrentHash = SHA256(d_list).

Step 4: Return: Output the CurrentHash.

Algorithm 2: Encryption and Decryption (PBE with MD5 & DES)

Password-Based Encryption (PBE) is a symmetric encryption method that uses a password or key for both encryption and decryption. This algorithm combines the MD5 and DES cryptographic techniques for secure data processing.

KeyGen(M): Generate a probabilistic key K from
the data M.

Enc(K, M): Encrypt the data M with the symmetric key K to produce ciphertext C_data.

Dec(K, C): Decrypt the ciphertext C using the same key K to retrieve the original data M.

Steps:

1. Key Generation: KeyGen(M) generates a key K for encryption based on input data M.
2. Encryption: Enc(K, M) applies the symmetric key K to the data M and produces the encrypted output C_data.
3. Decryption: Dec(K, C) decrypts the ciphertext C using the same key K and recovers the original data M.

Algorithm 3: Role-Based Access Control (RBAC)

This algorithm enforces a role-based access control mechanism, which ensures that access to files is granted based on user roles and attributes like email and file access permissions.

Input: Attribute Email-ID, File Data, File Key Data.

Output: Access policies or user permissions for file sharing.

Step 1: Initialize the data string S_list[] and set variables a = 0, k = 0, User Email-ID.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304128

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Step 2: Read the File Data and File Key Data.

Set a ← {filekey list [i . . . n]}.

Set k ← {Email-ID list [i . . . n]}.

Step 3: For each a in S_list[], do:

If key_data.Equals(a) and User Email-ID.Equals(k), then display the file sharing information for the user.

Otherwise, show the message indicating the user does not have file sharing access.

Step 4: End the procedure.

IV. RESULT AND ANALYSIS

Accurate matrices measurement is critical for evaluating process efficiency in cloud-based systems. The app under evaluation is hosted on an Amazon EC2 public cloud console with the following configuration: Processor: Intel 2.8 GHz i3 , Architecture: Java 3-tier architecture platform and RAM: 4 GB For the system evaluation, two physical network devices with Wi-Fi and ten virtual machines on Amazon EC2 were used as part of the public cloud platform setup. After implementing key parts of the system, we were able to achieve satisfactory output performance. The following table (Table 1) shows the system performance metrics, including the encryption and decryption times for the proposed algorithms, PBEWithMD5AndDES and SHA-256, for plain text conversion and encryption/decryption processes

Table 1: System performance (Estimated)

File Data Size in KB	Encryption time (Milliseconds)		Decryption time (Milliseconds)	
	Existing	Proposed	Existing	Proposed
5	495	425	624	612
10	620	526	745	733
15	940	890	954	810
20	1060	995	975	890

Figure 2 below illustrates the results of various deduplication techniques. It calculates several key metrics, including file size, deduplication ratio, and processing time for both fixed size and variable size methods.

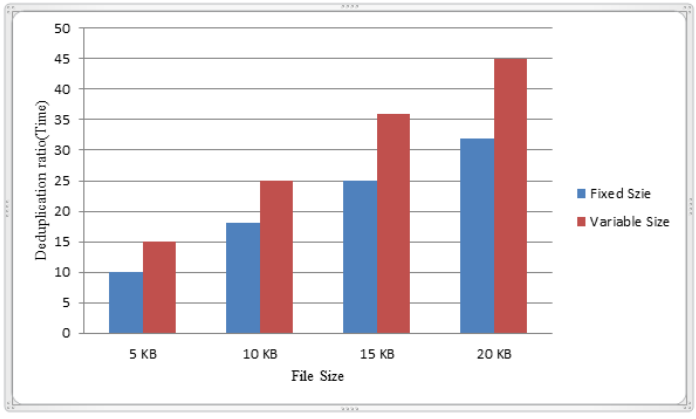


Figure 2. Comparison Analysis of fixed and Variable Size

The deduplication ratio is used as a key indicator of the effectiveness of the deduplication process. The following results demonstrate the impact of deduplication techniques on smaller and larger datasets. For the smaller files, the evaluation was conducted using files of 5 KB, 10 KB, 15 KB, and 20 KB. While these smaller file sizes were useful for testing, larger datasets are expected to show even higher levels of deduplication success.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. CONCLUSION

The project proposes a secure and efficient de-duplication process for cloud storage systems, leveraging advanced cryptographic techniques. The system ensures both data confidentiality and integrity by implementing authorization mechanisms before performing duplication checks. This approach improves data security by examining user privileges rather than relying solely on data characteristics for duplication checks. The security overhead is minimal compared to traditional methods, making the proposed solution an effective strategy for cloud-based storage systems.

REFERENCES

[1] B. M. Salih and O. K. Jasim Mohammad, "Cloud Data Leakage, Security, Privacy Issues and Challenges: Review," Procedia Comput Sci, vol. 242, pp. 592–601, 2024, doi: 10.1016/j.procs.2024.08.113.
[2] M. M. Belal and D. M. Sundaram, "Comprehensive review on intelligent security defences in cloud: Taxonomy, security issues, ML/DL techniques, challenges and future trends," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, pp. 9102–9131, Nov. 2022, doi: 10.1016/j.jksuci.2022.08.035.
[3] Z. R. Alashhab, M. Anbar, M. M. Singh, I. H. Hasbullah, P. Jain, and T. A. Al-Amiedy, "Distributed Denial of Service Attacks against Cloud Computing Environment: Survey, Issues, Challenges and Coherent Taxonomy," Applied Sciences, vol. 12, no. 23, p. 12441, Dec. 2022, doi: 10.3390/app122312441.
[4] T. Alyas et al., "Multi-Cloud Integration Security Framework Using Honeypots," Mobile Information Systems, vol. 2022, pp. 1–13, Aug. 2022, doi: 10.1155/2022/2600712.
[5] H. S. Al-Qahtani, "A Taxonomy of Factors that Influence the Multiple-Cloud Computing Utilization," in 2024 13th International Conference on Computer Technologies and Development (TechDev), IEEE, Oct. 2024, pp. 91–95. doi: 10.1109/TechDev64369.2024.00024.
[6] Srujan Reddy Anugu, "Optimizing Data Flow in Multi-Cloud Environments: A Technical Deep Dive," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 11, no. 1, pp. 1466–1473, Feb. 2025, doi: 10.32628/CSEIT251112154.
[7] K. J. Merseedi and Dr. S. R. M. Zeebaree, "Cloud Architectures for Distributed Multi-Cloud Computing: A Review of Hybrid and Federated Cloud Environment," Indonesian Journal of Computer Science, vol. 13, no. 2, Apr. 2024, doi: 10.33022/ijcs.v13i2.3811.
[8] C. T. D. Pravina, N. Arya, S. Sharma, K. Chakraborty, S. K. Rakesh, and A. K. Singh, "Image Based Security System in cloud resource Scheduling," in Proceedings of the 5th International Conference on Information Management & Machine Intelligence, New York, NY, USA: ACM, Nov. 2023, pp. 1–6. doi: 10.1145/3647444.3647861.
[9] P. R. V. -, "Securing Patient Data in Healthcare Cloud Systems: A Technical Overview," International Journal on Science and Technology, vol. 16, no. 1, Mar. 2025, doi: 10.71097/IJSAT.v16.i1.2754.
[10] E. Kamau, T. Myllynen, S. D. Mustapha, G. O. Babatunde, and A. A. Alabi, "A Conceptual Model for Real-Time Data Synchronization in Multi-Cloud Environments," International Journal of Multidisciplinary Research and Growth Evaluation, vol. 5, no. 1, pp. 1139–1150, 2024, doi: 10.54660/.IJMRGE.20247.5.1.1139-1150.
[11] K. J. Merseedi and Dr. S. R. M. Zeebaree, "Cloud Architectures for Distributed Multi-Cloud Computing: A Review of Hybrid and Federated Cloud Environment," Indonesian Journal of Computer Science, vol. 13, no. 2, Apr. 2024, doi: 10.33022/ijcs.v13i2.3811.
[12] Xue K, Xue Y, Hong J, Li W, Yue H, Wei DS, Hong P. RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage. IEEE Transactions on Information Forensics and Security. 2017 Apr;12(4):953-67.
[13] Kan Yang and Xiaohua Jia, Expressive, E_cient, and Revocable Data Access Control for Multi-Authority Cloud Storage, IEEE Transactions on parallel and distributed systems, VOL. 25, NO. 07, July 2014.
[14] Zhongma Zhu and Rui Jiang proposed A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud in IEEE TRANSACTIONS ON PAR- ALLEL AND DISTRIBUTED SYSTEMS, VOL. 27, NO. 1, JANUARY 2016.
[15] N. Attarpadung, B. Libert, and E. Pana_eu, Expressive keypolicy attribute based encryption with constant-size ciphertexts, in 2011.
[16] F. Zhang and K. Kim. ID-Based Blind Signature and Ring Signature from Pairings. In ASIACRYPT 2002, volume 2501 of Lecture Notes in Computer Science, pages 533547. Springer, 2002.
[17] J. Han, Q. Xu, and G. Chen. E_cient id-based threshold ring signature scheme. In EUC (2), pages 437442. IEEE Computer Society, 2008.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[18] J. Yu, R. Hao, F. Kong, X. Cheng, J. Fan, and Y. Chen. Forward secure identity based signature: Security notions and construction. Inf. Sci., 181(3):648660, 2011

[19] Yang K, Jia X. DAC-MACS: E_ective data access control for multi-authority cloud storage systems. InSecurity for Cloud Storage Systems 2014 (pp. 59-83). Springer, New York, NY.

[20] Guangyan Zhang at. al. proposed CaCo: An Efficient Cauchy Coding Approach for Cloud Storage Systems in IEEE Feb 2016.

[21] Ibrahim Adel Ibrahim at. al. proposed Intelligent Data Placement Mechanism for Replicas Distribution in Cloud Storage Systems in 2016 IEEE International Conference on Smart Cloud.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details