RESEARCH PAPER

A Lightweight Hash Oriented Model for Verifiable Data Deduplication in Cloud Environments

Nivashini C[1], Dr. K P Parthiban[2], Tharun S[3], Vignesh S[4], Naveen S[5]

[1]Assistant Professor, Department of Electronics and communication Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.
[2]Assistant Professor, Department of Electrical and Electronics Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India.
[3,4,5]UG Scholar, Department of Computer Science and Engineering, V.S.B College of Engineering Technical campus, kinathukadavu, Coimbatore, Tamilnadu, India
[1*]Email: mailmenivash@gmail.com [2]Email: drparthibankp@gmail.com

Abstract—Cloud storage systems rely on data deduplication, a process that eliminates duplicate copies of data to minimize storage expenses. This paper proposes attribute-based access control (ABAC) as a relevant access control mechanism to address the data integrity and confidentiality challenges in cloud storage. Such a service provider can tamper with stored data and duplication verification results, hence overcharging clients. To overcome on these issues, this project proposes a Lightweight Hash Oriented Model for Verifiable Data Deduplication in Cloud Environments. The verification mechanism is built on a hash-based integrity verification scheme and a secure, user-initiated duplication checking scheme to ensure data integrity and duplication accuracy. It provides an option for the users to create more flexible verification tags while still maintaining deduplication, enabling the verification of integrity independently from downloading the whole file. Finally, a secure challenge–response mechanism has been established to ensure that the CSP does not behave dishonestly when carrying out the duplication checking procedures. By combining encryption, tag management and lightweight verification methods, the model reduces overheads on computation, storage and communication. Security analysis and performance evaluation prove that the proposed approach can resist data tampering and maintain high efficiency and practicality in real-world Cloud environments, as well as resist dishonest duplication reporting.

Keywords—Cloud Storage, Data Deduplication, Hash-Based Verification, Attribute-Based Access Control, Data Integrity, Security.

How to cite this article: Nivashini C, Parthiban KP, Tharun S, Vignesh S, Naveen S. A Lightweight Hash Oriented Model for Verifiable Data Deduplication in Cloud Environments. Int J Drug Deliv Technol. 2026;16(76s): 1070-1077. DOI: 10.25258/ijddt.16.76s.94

INTRODUCTION

One of the most popularly used technologies for the storage and the management of large quantities of digital information is by means of cloud computing. Cloud storage services can provide organizations and individuals with solutions for remotely storing and accessing information that are scalable, flexible, and cost-effective. With the enormous growth of digital services/applications has come an increasingly large amount of data stored in, and generated from, cloud service environments [1] [6]. Because of this growth in data, cloud computing service providers are now presented with significant challenges in efficiently managing their storage resources and ensuring that the data they store is secure and meets reliability standards.

To help solve the problem associated with redundant data storage, many cloud computing systems use data same file, then the cloud service provider can use the cryptographic hash value associated with each file to identify the duplicate data and prevent storing redundant copies of the same file, which will reduce the total amount of storage used by the cloud service provider and increase the overall efficiency of the system [4] [9]. Several very recent studies have indicated that the storage utilization and operational costs are significantly improved for large scale cloud computing infrastructure as a result of deduplication practices [10] [11].

However, while the use of traditional data deduplication techniques has some clear benefits, they also have a few very critical security liabilities introduced into the cloud service environment. As the cloud service providers host and manage the physical storage environments that contain the user's data, the user has to trust the cloud service provider to perform the data deduplication process accurately and trustworthily. Unfortunately, this may expose users to data integrity violations and inaccurate billing by the cloud service provider for their use of the storage service

[2] [12]. Therefore, the cloud computing community must develop methods and procedures to reliably verify the accuracy and trustworthiness of the cloud computing service providers' data deduplication processes so that

trust can continue to exist between cloud users and their cloud service providers.

Another key area of concern for cloud storage systems is ensuring that strong access control and data confidentiality are maintained. Multiple users often share access to the same stored data resources in Cloud storage systems, which increases the risk that someone will access data in an unapproved manner. The attribute-based access control methods commonly suggested for providing fine-grain access policies to comply to an organization's policies based on user's attributes [7] [14] thereby allowing data owners to create access policies that provide data access only to those users who are authorized to access sensitive data stored in the Cloud.

In addition to implementing access control, the integrity verification of outsourced data is a requirement of critical importance. Integrity verification methods allow users to determine if the information they have stored in the cloud service provider has been changed or corrupted by the provider.

management, and verification methods to improve security of data stored in Cloud storage systems. [5] [6] Conference level research has also proposed several challenge-response verification protocols and hashing based deduplication detection methods to improve the reliability of deduplication actions. [15] [17] [19] However, current solutions still require significant computational overhead and low scalability due to their involvement of cryptographic computations.

As a result, there needs to be an efficient and low overhead verification methods for duplicates, with verification that duplicates have not been corrupted and with verification that the duplicates are truly duplicates of the "real" data without any computational overhead; therefore, the Lightweight Hashing Model (a system that provides verification for duplicate verification in cloud environments) has been created. This new model utilizes integrity verification based on hashing, provides access restrictions based on attributes and a secure challenge-response methodology in order to verify duplicates safely and to protect against the fraudulent actions of cloud services. Furthermore, by utilizing lightweight hashing methods, encryption and flexible tagging for verification purposes, the Lightweight Hashing Model is an innovation that increases the amount of data that can be stored safely and with decreased costs associated with storage.

Finally, the rest of this document is organized in the following manner. Section two will discuss other research related to secure data deduplication and techniques used to verify cloud storage. Section three provides a description of a proposed lightweight hash-based model and system architecture using hashes to allow for the verification of deduplicated data in a cloud environment. Section four provides details on how the system will be implemented, including hash-based methods of verification and access control. Performance evaluations and results for the new approach will be provided in section five. Finally, section six will summarize the content of the entire paper and suggest potential directions for future research aimed at improving the security of deduplication systems within cloud computing environments.

I. RELATED WORKS

The rapid growth of cloud services has created new challenges for researchers to provide storage efficiency while also implementing adequate data security measures. Researchers have also combined a number of different techniques to accomplish both secure and efficient deduplication in cloud-based environments through a combination of encryption, integrity verification and access control mechanisms.

Zhang et al. [1] developed a secure efficient deduplication framework that identifies duplicate data blocks within the cloud at scale and optimizes use of stored data by using cryptographic hashing to identify duplicate data blocks while maintaining confidentiality of original plaintext throughout storage. Li et al. [2] created a cloud storage auditing mechanism to maintain user privacy through secure methods for verification while using deduplication techniques, with their primary focus being on maintaining integrity of data stored in a cloud with a means to identify duplicate records.Kumar et al. [3] developed a method for integrity verification on cloud storage that allows users to check for integrity of stored data without downloading the whole file. This helps minimize communication overhead. Chen et al.

[4] offered an efficient deduplication framework which combines secure verification methods with optimized storage management techniques, providing improved data reliability and enhanced efficiency when deduplicating distributed cloud storage.

Al-Zahrani et al. [5] proposed a hash-based secure deduplication method with both integrity verification and encryption capabilities. Their results demonstrate an effective reduction of redundancy while providing data security using hash- based deduplication methods. Likewise, Sun et al. [6] provided a method for preserving user privacy through deduplication that still allows for effective user operation on their data.

There has been considerable interest in investigating attribute based access control systems as they pertain to the field of secure cloud storage. Mishra et al. [7] have developed a secure deduplication model that incorporates an attribute based access control mechanism to enforce fine grained access policies to data. Their proposed system permits a cloud environment the ability to limit access to sensitive information based on the account holder's specific attributes/roles. Huang et al. [8] further study

secure verification and improved upon it by developing lightweight cryptographic protocols; these protocols provide an efficient means for data integrity verification while reducing computational overhead.

Sathyanarayanan et al. [9] have proposed creating a secure file management system that improves mechanisms for secure storage and accesscontrols in distributed environments using an encrypted file management system. Jino et al. [10] present a deep-learning model that can be used to detect and mitigate cyber-attacks on network systems which helps improve the reliability and overall security of the system. Overall, these studies show the importance of implementing strong security mechanisms in today's cloud storage systems.

Scalable deduplication architectures have also been explored. Singh et al. [11] developed a scalable deduplication method that improved upon the performance of storage within cloud environments by optimizing the performance of duplicate detection algorithms. Liu et al. [12] developed a verification method that gives users the ability to independently verify that the cloud provider correctly executed the deduplication operations.

Roy et al. [13] published the first known lightweight verification protocol based on hash-based challenges to detect possible data manipulation within cloud storage systems.

The research on this topic has also been supported by studies published in conference proceedings. Kumar et al. [15] introduced a secure cloud storage model that integrates deduplication with integrity verification. Hassan et al. [16] proposed a lightweight, secure deduplication framework that maximizes performance while ensuring data integrity. Gupta et al. [17] offered a challenge-response mechanism to verify the integrity of the data stored on the cloud and to prevent the cloud service provider from acting in a manner that would be considered dishonest during the process of duplicating files for the purpose of duplicating.

Nair et al. [18] published an attribute-based secure deduplication framework aimed at strengthening access control mechanisms in cloud storage systems, while Wang et al. [19] presented a hash-based deduplication method designed to boost the efficiency of duplication detection while ensuring the integrity of the associated datasets. Finally, Chen et al. [20] developed a secure method for detecting duplication that uses lightweight cryptographic techniques to enhance the reliability of the system.

Park et al. [21] developed an integrity verification model specifically for deduplicated cloud data. The model is based on using verification tags and hash value verification of data which guarantees that the data stored will not change. Verma et al. [22] also described a secure deduplication method that incorporates efficient key management techniques in an effort to increase the security of the system while minimizing the amount of computational complexity associated with the method.

These studies have really improved the security and efficiency of cloud deduplication systems. However, a lot of existing methods still use cryptography that increases the workload on computers and the costs of communication. Cloud deduplication systems need an efficient way to verify that data is duplicated correctly while keeping the data safe. The proposed Lightweight Hash-Oriented Model for Verifiable Data Deduplication in Cloud Environments is a solution to these problems. It uses hash-based verification to check the integrity of cloud deduplication systems. It also uses attribute-based access control and a secure challenge–response protocol. This makes cloud deduplication systems more secure and efficient. The Lightweight Hash-Oriented Model for Verifiable Data Deduplication, in Cloud Environments improves the security and storage of cloud systems.

II. PROPOSED METHODOLOGY

A. Overview

In Cloud Environments, the proposed system provides a Lightweight Hash-Based Model for Verifiable Data Deduplication to enhance storage efficiency while ensuring integrity and secure verification for duplicated data. Cloud storage typically has large amounts of redundantly stored data, which results in increased storage expenditure and computational load. Typical data deduplication methods remove previously stored duplicate files; however, existing deduplication technologies do not offer any means for users to verify that their service provider is accurately performing duplicate checks.

The proposed model integrates any of the following mechanisms: (1) integrity verification through hash values;

(2) attribute-based access control (ABAC); and (3) challenge response-based verification to provide verification capability. To achieve the objectives of this system, it allows for more efficient identification of duplicate data and allows users to check the integrity of file duplication without requiring the user to download their entire file.

As part of this process, whenever a user uploads a file to the cloud storage system, a secure hash based on the uploaded file data is generated (e.g., SHA-256). This hash willuniquely represent the file's contents. The hash that was generated will be compared to the hashes that are listed in the cloud storage system's database to determine if the file uploaded to the cloud storage system already exists in the cloud storage system.

When the created hash is the same as an already-

stored hash value, the file is identified as being a duplicate, and thus no additional copy of the file is created. A reference pointer is created pointing to the already-stored data instead of creating an additional copy. If the hash does not appear in the database for any reason, the file is then treated as being a new data block and will be stored with the cloud storage system.

To ensure that the stored data is not compromised and thus protect against dishonest actions from the cloud service provider, the system creates verification tags for all files that are stored. The purpose of these verification tags is to allow users to check the validity of their data, without having to retrieve it, using a challenge-response method that allows the user to verify the duplication-checking function of their cloud service provider and that the data will still be valid based upon the cloud service provider keeping the integrity of the data stored.

This combination of lightweight hashing, verification tags, and secure access control systems will help minimize redundant storage in the cloud, while still allowing users to have very high levels of security and very efficient verification functions.

B. System Architecture

The system architecture of the proposed Lightweight Hash-Oriented Model for Verifiable Data Deduplication in Cloud Environments is made up of multiple functional modules that combine to allow secure, efficient, and verifiable storage of data in a cloud. Each of these functions interacts with one another to perform the necessary halves of duplicate detection, secure storage of data, and verification of the integrity of the stored data, as illustrated below in the Fig 1.

Starting with the End User (Data Owner) uploading a file to cloud storage system, the first step is an Authorization check performed by the Attribute Validator module. This module is responsible for enforcing the policies of Attribute-Based Access Control (ABAC) and ensuring that the user has permission to upload or access data on the cloud in accordance with ABAC policy. Thus, this module ensures that only correctly authorized users may create an account or access an existing account of a cloud storage system.

Once the file has been successfully validated, it is passed to the Lightweight Hash Generation Module where a secure cryptographic hash value using SHA-based hashing algorithms is generated. The hash value generated from the file will serve as a unique identifier (or fingerprint) of that file and will be used to determine duplicates within the cloud storage system.

After generating the hash value, it will be sent to the Deduplication Checker. The Deduplication Checker will compare the hash value against the hashes already existing in the cloud database to see if the file being uploaded already exists. If the Deduplication Checker finds a hash-match then the file will be marked as duplicate. In this case,

a pointer will be updated to point to the existing data rather than to store another copy of the file.

If the Deduplication Checker does not find a matching hash for the file that was uploaded, then the file will be considered as unique. The file will be sent to the Secure Data Storage Module, where it will be stored in the cloud database. Because of this process, duplicate data storage is significantly reduced and storage efficiency is improved in cloud environments at scale.The system also has an IntegrityVerifier module that re-verifies each stored file by periodically performing a hash re-check on a file to ensure that the file stored on the cloud has not been altered or changed while being stored on the cloud.

Lastly, the Audit and Log Monitoring Module keeps logs of all activities performed in the system, including files uploaded into the system, deduplication decisions made by the Deduplication Management Module, and verification operations performed by the Verification Module. It tracks system activities to reveal any security breaches caused by any malicious act from the cloud service provider.

Therefore, the proposed architecture combines access control, lightweight hashing, deduplication, secure storage, and verification algorithms for efficient and secure cloud data management.

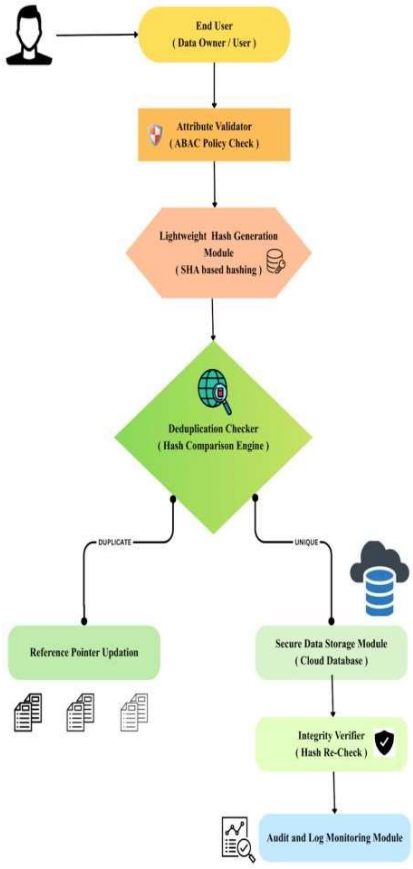

Fig. 1. System architecture of the proposed lightweight hash-oriented model for verifiable data deduplication in cloud environments.*Algorithm*

The proposed system performs secure and verifiable data deduplication based on a lightweight hash-based mechanism. Data integrity and prevention of dishonest behavior from the cloud service provider are also secured using the lightweight hash-based method. In this regard, the deduplication process entails four phases, which include hash generation, duplication checking, secure storage, and integrity verification.

Input: File F uploaded by the user.
Output: Deduplication decision and verification status.

1. The end user uploads file F to the cloud storage system.
2. The system checks the user to see if they are allowed to do this.
3. It does this through the Attribute Validator, which's like a security check to make sure the user is who they say they are.
4. If the user is allowed the file F goes to the step.
5. The system looks at file F. Makes sure it is okay to proceed.
6. Now the system makes a code called a hash value for file F.
7. This code is like a unique name for file F.
8. Generate a cryptographic hash value H=SHA256(F).
9. The system sends this code H to the Deduplication Checker.
10. This checker looks at code H and compares it to codes it already has.
11. If code H is the same as a code the system already has. It means file F is a duplicate. The system then updates a link that shows the user has file F.
12. It does not store file F again because it already has a copy.
13. If code H is not the same, as any code the system has.
14. The system stores file F in a place called the Secure Cloud Storage Module.
15. Then it makes a tag to check if file F is still okay.
16. Perform Integrity Verification using hash re-checking.
17. Record system activity in the Audit and Log Monitoring Module.
18. Finally, the system tells the user what happened to file F.
19. It says if file F is a duplicate or not and if it is safe.

III. PERFORMANCE ANALYSIS

The performance evaluation of the proposed Lightweight Hash-Oriented Model for Verifiable Data Deduplication in Cloud Environment is based on many important metrics such as storage efficiency, computational overhead and security effectiveness. The above metrics are used to analyse their efficiency in reducing redundant data while maintaining data integrity and secure access control.

A. Storage Efficiency

One of the primary targeted goals of data deduplication is to reduce redundant data within the cloud storage system. In the proposed model, duplicate files were identified through cryptographic hash values using a lightweight hash value generation module. In the event that a duplicate file is

detected, a copy of the duplicate file is not stored, instead, a pointer reference is created linking a user to the stored file. This mechanism majorly mitigates storage consumptions in a cloud environment. This mechanism allows for significant reduction of a total amount of storage consumption in the cloud environment, so unlike in traditional storage systems, when every uploaded file is stored separately, the proposed model guarantees that only unique data are stored. As a result, this system achieves a high degree of storage utilization and leads to reduced storage costs by cloud service providers.

B. Computational Overhead

The new system has two main types of computational overhead (computing resources required to run the solution); these are the hash value generation and script to identify duplicate files. When a new file is uploaded, a secure hashing algorithm (specifically SHA-256) generates a secure hash value for every new file, which provides a unique representation of that files content. The generation of this hash value takes very little time to compute, meaning that as the user uploads more and more files the system will remain able to scale without causing prolonged delays in processing times.

During the deduplication process, the system will create and compare all hash values against those that are already stored in the system in order to determine if there are any files that are identical to files that have already been uploaded to the system. The benefit to hashing is that entire files do not need to be compared, but rather only their hash values. As a result, without needing to compare entire files, the resulting processing times improve vastly (much faster) for duplicate detection, plus reduced amount of work for the system when performing duplicate detection.

In conclusion, the proposed system has been designed in a way that allows for efficient identification of duplicate files with minimal computing costs. The proposed system will provide the user with high performance and scalability by the use of lightweight hashing and optimized techniques to compare the hash values.

C. Security Analysis

An integrated set of safeguards is used to maintain the confidentiality and integrity of the data so that it can be protected from unauthorized access and modification by anyone outside of your organization. The ABAC model allows only authorized users to upload or access files that are stored in the cloud, thereby limiting the exposure of sensitive information to unauthorized individuals. When a hash value is calculated for the original file, if an unauthorized action occurs that changes the original version of the file, then using any type of modification to change the original version value will also produce a new hash value for the file. This provides a mechanism for any user who is accessing the cloud storage system to confirm that no unauthorized modification occurred in the database. The challenge/response verification also allows for the verification of any potential issues that may arise when a cloud service provider is attempting to deceive its customers while they are performing their duplication-checking duties. Overall, these shared methodologies provide a higher level of trust and reliability for users of cloud storage solutions.

Table I. Performance comparison of the proposed model with existing systems

Metric	**Traditional Storage**	**Existing Deduplication**	**Proposed Model**
Storage Usage	High	Medium	Low
Duplication Detection	No	Yes	Yes
Integrity Verification	No	Limited	Strong
Computation Overhead	High	Medium	Low

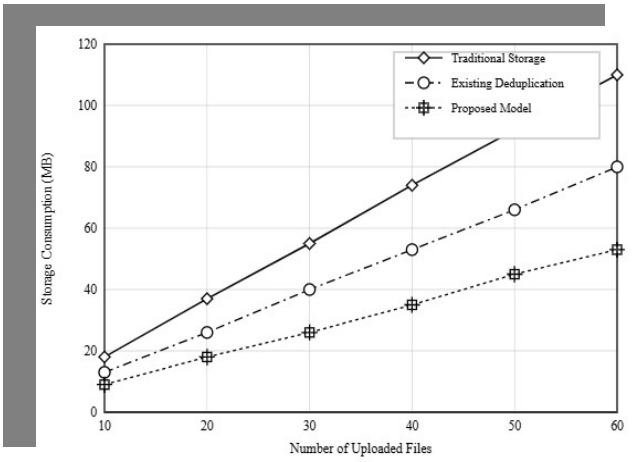

Fig. 2. Storage consumption comparison with respect to number of uploaded files.

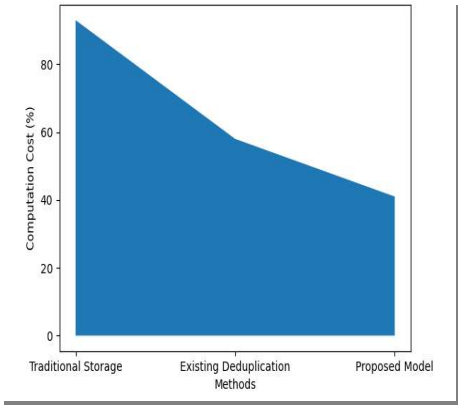

Fig. 3. Computational overhead comparison of traditional storage systems and the proposed deduplication model.

An assessment of how well the lightweight hashing-based deduplication model performed on both data storage requirements and the amount of time it requires to store the data will be summarized here. Fig. 2 and Fig. 3 are included in order to show that the new lightweight hashing approach uses less space for storing duplicated items than previous methods do, and that it requires less CPU time (and therefore uses less CPU power) than traditional methods or existing large record deduplication methods.CONCLUSION

This paper presents a lightweight hashing model for deduplicating data in a cloud environment and managing issues related to redundant storage, how access control is accomplished, and how integrity can be verified. In this approach, cryptographic hashing methods are used to identify duplicate data and thereby eliminate redundant storage and increase storage efficiency. The addition of a mechanism for controlling access to stored data through an attribute-based access control (ABAC) approach and a mechanism for verifying the integrity of the stored data creates an overall solution that provides a significant advantage over existing systems for verifying data integrity, reducing the cost of storage, and requiring less computing power to produce results. The proposed solution also demonstrates significant improvements over traditional storage methods and current solutions for verifying data integrity; therefore, it has the potential to serve as a secure and efficient means for managing cloud-based storage resources in large-scale systems. In addition, future research may include additional security enhancements to verify the integrity of stored data through research on using biometrics for authentication, using blockchain technology for the verification of data and implementing techniques to optimize storage through the use of "intelligent" techniques.

REFERENCES

[1] Y. Zhang, J. Chen, and X. Li, "Secure and efficient data deduplication in cloud storage systems," *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 1102–1114, 2023.

[2] H. Li, X. Liu, and K. Ren, "Privacy-preserving cloud storage auditing with secure deduplication," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 410–423, 2023.

[3] S. Gupta and P. Kumar, "Lightweight integrity verification for secure cloud storage," *IEEE Access*, vol. 11, pp. 52540–52552, 2023.

[4] Z. Chen, J. Xu, and Y. Wang, "Efficient deduplication framework with secure verification for cloud environments," *Future Generation Computer Systems*, vol. 140, pp. 42–53, 2023.

[5] M. Al-Zahrani and F. Alruwaili, "Secure cloud storage using hash-based data deduplication and integrity verification," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–15, 2023.

[6] L. Sun and Q. Wang, "Efficient and privacy-preserving deduplication for cloud storage systems," *IEEE Access*, vol. 10, pp. 77432–77444, 2022.

[7] A. Mishra and R. Singh, "Attribute-based secure deduplication scheme for cloud storage," *Journal of Network and Computer Applications*, vol. 208, pp. 103501, 2022.

[8] T. Huang and S. Li, "Lightweight cryptographic verification for secure cloud storage systems," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3521–3533, 2022.

[9] M. Sathyanarayanan, K. Manivannan, K. Mithin, G. Manoj, M. Mugesh and N. Kishore, "Design and Implementation of a Cybersecurity Framework for Encrypted File Management System," in *2025 6th International Conference for Emerging Technology (INCET)*, IEEE, 2025.

[10] R. F. Jino, A. M. Paulsamy, G. Shanmugam and R. K. Vishwakarma, "Enhancing network security: a deep learning-based method to detect and diminish attacks," *International Journal of Electronic Security and Digital Forensics*, vol. 17, no. 5, pp. 631–645, 2025.

[11] D. Singh and V. Sharma, "Secure and scalable deduplication techniques in cloud computing," *Journal of Systems Architecture*, vol. 128, 102515, 2022.

[12] Y. Liu and J. Li, "Efficient verification schemes for deduplicated cloud storage systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 3105–3117, 2023.P. Roy and S. Bhattacharya, "Hash-based lightweight verification protocol for cloud storage," *IEEE Access*, vol. 11, pp. 89721–89733, 2023.

[13] R. Sharma and K. Gupta, "Secure deduplication model with attribute-based access control in cloud storage," *Future Generation Computer Systems*, vol. 146, pp. 154–165, 2024.

[14] A. Kumar and S. Verma, "Secure cloud storage with deduplication and integrity verification," in *Proc. IEEE International Conference on Cloud Computing*, 2023, pp. 321–326.

[15] M. Hassan and L. Zhao, "Lightweight secure deduplication framework for cloud environments," in *Proc. IEEE International Conference on Big Data*, 2022, pp. 1875–1880.

[16] R. Gupta and A. Patel, "Efficient challenge-response verification for cloud deduplication," in *Proc. IEEE International Conference on Communications*, 2023, pp. 4451–4456.

[17] S. Nair and P. Krishnan, "Secure attribute-based deduplication in cloud storage," in *Proc. IEEE International Conference on Information Security*, 2022, pp. 220–225.

[18] T. Wang and Y. Zhao, "Hash-based deduplication with integrity verification for cloud systems," in *Proc. IEEE International Conference on Cloud Engineering*, 2021, pp. 210–215.

[19] J. Chen and X. Zhang, "Secure duplication detection using lightweight cryptographic verification," in *Proc. IEEE International Conference on Distributed Computing Systems Workshops*, 2021, pp. 92–97.

[20] J. Park and H. Kim, "Integrity verification model for deduplicated cloud data," *IEEE Access*, vol. 10, pp. 12451–12463, 2022.

[21] K. Verma and S. Patel, "Secure deduplication with efficient key management for cloud storage," *Computer Networks*, vol. 207, 108828, 2022.