THE SCIENTIFIC TEMPER (2026) Vol. 17 (5): 6213-6222

E-ISSN: 2231-6396, ISSN: 0976-8653

Doi: 10.58414/SCIENTIFICTEMPER.2026.17.5.01

https://scientifictemper.com/

RESEARCH ARTICLE

A Secure Ownership Verification Model for Accessing Deduplicated Cloud Data

Dr. G. Uma*

Abstract

Cloud storage platforms commonly use data deduplication to remove repeated file copies and improve storage utilization. While this approach helps reduce storage overhead and network usage, it raises an important security concern: how to confirm that a user requesting access to a deduplicated file is the rightful owner. To address this issue, this paper presents a secure ownership verification model for controlled access to deduplicated cloud data. The proposed method improves access validation by creating a file-specific verification value using both user-related and file-related attributes, such as user credentials, token details, and file tag information. This value is generated during file upload and recomputed whenever a user requests access to the stored file. Access is permitted only when the regenerated value matches the previously stored verification record, thereby preventing unauthorized retrieval. The model was implemented in a cloud environment using C#.NET and deployed on Microsoft Azure, and its performance was evaluated in terms of file access computation time for different file sizes. Experimental results show that the proposed approach outperforms existing methods such as CAE and Dekey by achieving lower computational time while also providing stronger ownership-based access control.

Keywords: Deduplicated cloud storage, Secure ownership verification, Access control, Credential-based validation, Cloud data security, Authorized file access, Storage optimization.

Introduction

Cloud computing has transformed the way individuals and organizations store, manage, and retrieve digital information. As cloud usage continues to expand, service providers must handle enormous volumes of repetitive and fast-growing data. To manage this challenge efficiently, cloud storage systems increasingly rely on data deduplication, a technique that removes duplicate copies of identical content and stores only a single instance of a file while maintaining logical references for authorized users (Prajapati P. et al., 2022).

Associate Professor, Department of Computer Science, Srimad Andavan arts & Science College (Autonomous), Affiliated to Bharathidasan University, Trichy.

***Corresponding Author:** Dr. G. Uma, Associate Professor, Department of Computer Science, Srimad Andavan arts & Science College (Autonomous), Affiliated to Bharathidasan University, Trichy. E-Mail: drumagphd@gmail.com

How to cite this article: Uma, G. (2026). A Secure Ownership Verification Model for Accessing Deduplicated Cloud Data. The Scientific Temper, **17**(5):6213-6222.

Doi: 10.58414/SCIENTIFICTEMPER.2026.17.5.01

Source of support: Nil

Conflict of interest: None.

Although deduplication offers clear operational and economic benefits, it also introduces important security concerns. When multiple users may point to the same stored object, the cloud system must carefully determine whether a user requesting access to that object is a legitimate owner or an unauthorized party attempting to exploit the deduplication mechanism (Lee M. et al., 2023). To address this challenge, this paper proposes a secure ownership verification model that strengthens ownership validation during file retrieval by combining user-specific credentials with file-specific . In the paper, this value is formed from the user identity, password, token, and tag associated with the file, and it is stored when the file is initially uploaded. When a user later requests access to the stored data, the system regenerates the verification value using the submitted credentials and the stored file metadata, and then compares the new value with the original record. Access is granted only if both values match. This design introduces a stronger link between user authentication and data ownership validation, thereby reducing the chance of unauthorized file retrieval in a deduplicated environment.

Another important feature of the proposed model is that it operates as part of a broader deduplication workflow rather than as an isolated authentication check. The paper explains that the cloud-based application integrates convergent encryption, key generation, token management,

© The Scientific Temper. 2026

Received: 12/03/2026 **Accepted:** 18/04/2026 **Published:** 22/05/2026

tag generation, and ownership verification into a unified implementation. This integration is important because secure deduplication is not achieved by storage reduction alone; it also requires coordinated handling of encryption keys, file identifiers, verification records, and access decisions. By embedding ownership verification directly into the file access process, the proposed model improves trust in data retrieval without sacrificing the storage efficiency gained through deduplication.

Related Work

Recent research on secure cloud storage has given significant attention to data deduplication, since it helps reduce storage overhead and communication cost by eliminating redundant copies of identical data. However, as deduplication becomes more widely adopted, researchers have increasingly recognized that storage efficiency alone is not sufficient. Secure deduplicated storage also requires reliable ownership validation, controlled access mechanisms, auditing support, and resistance to unauthorized retrieval. This has led to the development of several proof-of-ownership and ownership-aware security models in recent years.

A major stream of recent work focuses on improving proof-of-ownership verification for deduplicated cloud systems. Dave et al. (2025) proposed a secure and efficient ownership-verification method in which the proof is generated from the entire file rather than from a weak partial representation. Their approach attempts to balance stronger security with lower overhead by requiring limited block access in each iteration, thereby reducing the risk that an adversary could falsely claim file ownership using incomplete knowledge. This work is important because it clearly shows that ownership verification must be designed as both a security mechanism and a performance-aware procedure.

The same research direction was extended further by Dave et al. (2026) in a later study on blind and bidirectional ownership verification. In that model, ownership checking is not treated as a one-sided process. Instead, the design allows both the user and the server side to participate in a stronger mutual verification structure. This shift is meaningful because it reflects a broader movement in the literature from simple upload verification toward trust-aware ownership validation in cloud systems where neither side should be assumed fully transparent by default.

Another emerging direction is the move from interactive protocols toward non-interactive proof-of-ownership models. Yang et al. (2024) explored this transition by asking how cloud systems can securely determine who truly owns a file without relying on repeated interaction between the client and the server. This line of work is especially relevant because interactive schemes may introduce extra communication rounds and protocol complexity. Non-interactive designs attempt to preserve verification strength while simplifying the overall access process, making them attractive for practical cloud environments.

Building on this trend, Zhou et al. (2025) introduced MiS-PoW, a mirror-selected non-interactive proof-of-ownership scheme for cloud storage. Their method derives synchronized challenge information from the transport session and combines it with a stratified sampling strategy to improve the reliability of ownership checking. The study is notable because it addresses a common weakness in traditional sampling-based approaches: if challenge positions are too predictable or too narrow, attackers with partial information may still exploit the system. By strengthening challenge selection and reducing interaction overhead, the work contributes to more robust and efficient ownership verification.

In another recent contribution, Zhou et al. (2026) proposed ZKNiS-PoW, a privacy-preserving non-interactive proof-of-ownership scheme based on zk-STARK principles. This work reflects the growing interest in combining ownership verification with privacy-preserving cryptographic techniques. Rather than exposing verification details directly, the model allows a user to prove file possession in a way that reduces disclosure of sensitive information. Although such approaches are powerful, they may also introduce additional cryptographic complexity. This makes them highly relevant for comparison, particularly when discussing the trade-off between strong privacy guarantees and implementation practicality.

Beyond proof-of-ownership alone, several researchers have argued that secure deduplication must be integrated with access control and key management. Tang et al. (2024) proposed a lightweight cloud deduplication scheme that combines efficient access control, key management, and hybrid-cloud design. Their work emphasizes that secure deduplication cannot be treated as a standalone storage optimization problem. Instead, the security of deduplicated data depends on how encryption keys are handled, how access privileges are enforced, and how the public and private cloud components coordinate. This perspective strongly supports the view that access to deduplicated content should be governed through ownership-aware and credential-linked validation models.

Another relevant area of research concerns auditing and ownership management after data upload. Wang et al. (2024) addressed this issue by proposing a secure auditing and deduplication scheme with efficient ownership management for cloud storage. Their study highlights that ownership is not only an upload-time concern; it remains important throughout the data lifecycle, particularly when users are revoked or when access rights change over time. The introduction of ownership management and lazy ciphertext updating in their work shows that secure access

to deduplicated data must also account for dynamic control after storage has already occurred.

Similar concerns appear in domain-specific storage systems. For example, Yu et al. (2024) studied cloud-based electronic medical records and proposed a framework that combines auditing, data sharing, deduplication, and privacy protection. Their work demonstrates that in sensitive application areas, deduplicated cloud storage must satisfy multiple requirements simultaneously, including confidentiality, integrity, controlled sharing, and private information protection. This reinforces the broader argument that efficient storage alone is insufficient unless accompanied by trusted access mechanisms.

Recent literature has also explored blockchain-assisted deduplication and auditing. Vijayakumar et al. (2025) proposed a blockchain-enabled framework that combines deduplication, auditing, and fair payment mechanisms, while Jin et al. developed a blockchain-based auditable deduplication scheme for multi-cloud storage. These studies reflect another important direction in the field: researchers are increasingly interested in making deduplicated storage not only efficient and secure, but also transparent, auditable, and accountable. Although blockchain-based designs can strengthen traceability and integrity assurance, they often involve additional architectural and computational complexity when compared with lighter credential-driven ownership verification models.

Despite the progress made in proof-of-ownership, auditing, and secure deduplication techniques, the literature still does not fully address a practical mechanism for verifying user legitimacy at the moment of file retrieval. Many existing studies concentrate on proof construction, privacy preservation, or cryptographic efficiency, but they often treat ownership verification as an isolated protocol rather than as an integrated access-control process. As a result, the connection between user identity, file-specific metadata, and real-time authorization remains weak in many deduplicated cloud storage models. Several advanced methods also rely on complex cryptographic structures or hybrid-cloud designs, which can make practical deployment more difficult in lightweight and real-world cloud environments. In addition, ownership-related protection is frequently emphasized during upload or auditing stages, while secure verification during actual access is less clearly addressed. This creates a security gap because multiple users may logically reference the same deduplicated object, increasing the risk of unauthorized retrieval. Therefore, there is a clear need for a simpler and more effective ownership-aware access verification approach. Such a method should directly bind user credentials with file-related parameters and validate them before releasing stored data. Addressing this gap is essential for strengthening authorization in deduplicated cloud storage. The present research work responds to this need by proposing a practical credential-linked verification model for secure data access.

Problem Statement

Cloud storage has become a core component of modern digital infrastructure because it provides scalable, flexible, and cost-efficient data management. At the same time, the massive growth of cloud applications has produced large volumes of duplicate data, making deduplication an important technique for saving storage space and reducing bandwidth use. However, this efficiency introduces a serious security concern. When multiple users are linked to a single stored copy of the same file, the system must accurately verify whether the person requesting access is the genuine owner or an unauthorized user attempting to exploit the deduplication process. In many existing systems, ownership validation still depends on lightweight proof methods based on file hashes, sampled blocks, or isolated cryptographic evidence. Although these methods may reduce overhead, they often fail to create a strong connection between the requesting user, the stored file, and the final access decision. This weakness becomes highly risky in today's threat environment, where attackers may take advantage of leaked metadata, exposed verification details, reused credentials, or gaps between authentication and authorization to make false ownership claims.

The problem becomes even more critical because deduplicated storage changes the traditional notion of file possession. Unlike conventional storage, where each user maintains a separate copy, deduplicated cloud storage allows several users to reference the same physical object, which demands much stronger precision in access control. If this verification is not robust, the mechanism intended to improve efficiency can instead open the door to unauthorized retrieval, privacy breaches, and loss of trust in cloud services. Although recent research has proposed stronger proof-of-ownership schemes, privacy-preserving methods, and advanced cryptographic models, many of these approaches focus more on proof generation than on practical authorization during real file access. Some also introduce architectural or cryptographic complexity that limits their suitability for lightweight cloud deployments. Therefore, there remains a clear need for a practical, secure, and efficient ownership-aware verification model that binds user credentials with file-specific metadata and validates both at retrieval time. Addressing this issue is essential to ensure that the benefits of deduplication do not come at the cost of weakened data protection.

Proposed Methodology

This paper proposes a secure ownership verification model designed to control access to deduplicated cloud data through a credential-linked validation process. The methodology is built on the principle that access to a

deduplicated file should not be granted merely because a duplicate object already exists in the cloud. Instead, the system must establish that the requesting user is genuinely associated with the stored file through a verification process that combines both user-related credentials and file-related descriptors. In the paper this methodology is integrated into the upload and download workflow of the deduplication framework and is used specifically during file retrieval to validate ownership before access is allowed.

The proposed methodology operates within a cloud-based deduplication environment in which users first register with the system and then log in using their assigned credentials. Once authenticated at the application level, users are allowed to upload and retrieve files. However, file retrieval involves an additional ownership validation stage. This second-layer verification is the core methodological contribution of the paper. It ensures that even if a file is already present in deduplicated storage, access is granted only after the requesting user passes a file-specific verification check.

At the time of file upload, the system generates a sequence of file-related parameters required for secure storage and future access validation. These include a token (TKN) and a tag (TG) associated with the uploaded file. After these values are produced, the framework computes a secure verification value, which acts as the ownership-linked control parameter for that file. This verification value is derived by applying a cryptographic digest function to the combination of the user ID, password, file token, and file tag. By combining identity information with file metadata, the methodology creates a stronger and more specific verification structure than approaches that depend only on a single file-derived digest. Figure 2 shows the block-level architecture of the proposed secure ownership verification model for deduplicated cloud storage.

Figure 1 presents the structural organization of the proposed secure ownership verification model. The process begins with the user, who interacts with the cloud system through the authentication unit and application interface. Within the verification framework, the token generation module and tag generation module produce the file-specific parameters required for verification. These values, together with user credentials, are processed by the verification value generator to create a secure ownership-linked verification value. The generated value is stored in the ownership verification repository along with relevant user and file metadata. The duplicate detection unit checks whether the uploaded file already exists in encrypted deduplicated storage, while the logical reference manager maintains user-to-file associations for duplicate content. During file retrieval, the access verification and file release unit recomputes the verification value, compares it with the stored record, and grants access only when the ownership condition is satisfied.

To support this methodology, the framework maintains three coordinated record sets within the key and token management environment. The first stores deduplication-related information, including the user identifier, file identifier, token, encryption-related keys, and file tag. The second stores ownership verification information, including the user identifier, file identifier, token, tag, and the generated verification value. The third stores registered user login details, particularly the user ID and password. These linked record sets enable the system to retrieve the appropriate data elements during both upload-time

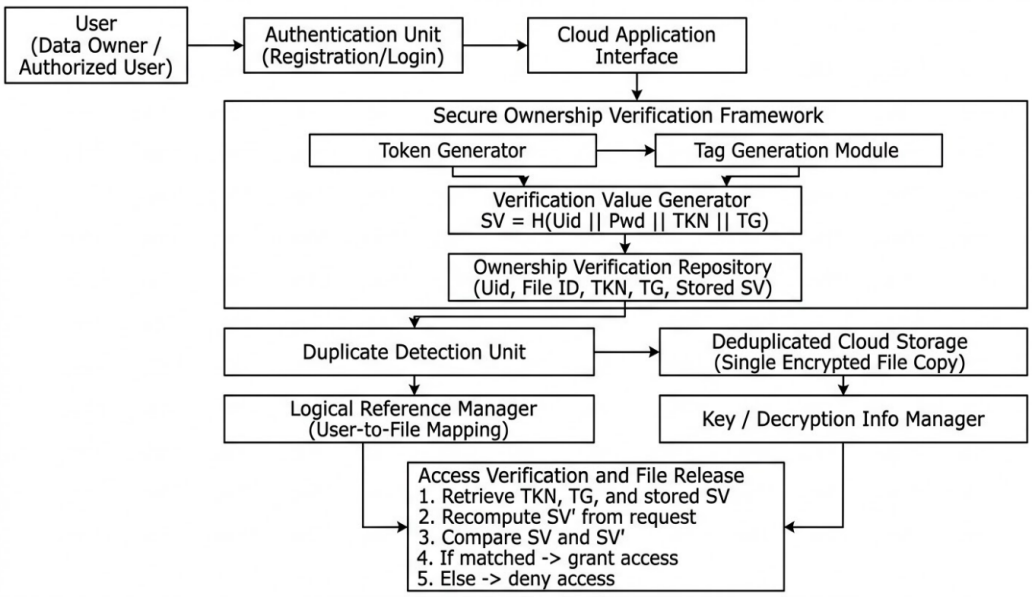


Figure 1: Block diagram of the proposed secure ownership verification model for deduplicated cloud data access

generation and access-time verification. The methodology therefore depends not only on a cryptographic operation, but also on structured metadata management across the deduplication framework.

A central part of the proposed methodology is the generation of the secure verification value during file upload. For each newly stored file, the system computes a digest-based value from the concatenation of the user ID, password, token, and tag. In conceptual form, the process may be represented as:

$$SV = H\left(Uid \,||\, Pwd \,||\, TKN \,||\, TG\right)$$

where:

- SV represents the secure verification value,
- Uid denotes the user identity,
- Pwd denotes the user password,
- TKN represents the token generated for the file,
- TG denotes the file tag, and
- H represents the cryptographic hash function used by the framework.

This verification value is created when the file is first uploaded and is then stored in the verification repository for later comparison. Because the value is tied to both the user and the file, it serves as an ownership-aware authorization marker rather than a general file digest.

The methodology also defines how the system behaves when a duplicate file is uploaded by another user. In such cases, the deduplication process checks whether the corresponding token and tag already exist in the framework. If the file is identified as a duplicate, the system does not create a fresh physical copy of the data. Instead, it creates a logical association for the new user to the existing stored content. Even in this case, the access process remains governed by the ownership verification mechanism. This is an important methodological point because it preserves storage efficiency while still requiring controlled and user-linked verification for access.

When a user later requests access to a stored file, the system requires the user to submit credentials again and identify the file to be retrieved. Using the user ID and file ID, the framework locates the relevant stored metadata, including the token, tag, and previously stored verification value. The system then regenerates a new verification value from the current user submission and the stored file descriptors using the same cryptographic process used at upload time. This newly generated value is compared with the verification value already stored in the system. Access is granted only if the two values match. If they do not match, the system treats the request as unauthorized and denies file access. This comparison-based design functions as a challenge-response style verification step embedded within the data access workflow.

Proposed Procedural Steps

The full methodology can be summarized as the following sequence:

- The user registers with the cloud deduplication system.
- The user logs in using valid credentials.
- During file upload, the framework generates the token and file tag.
- A secure verification value is computed from the user credentials and file metadata.
- The verification value is stored in the ownership verification repository.
- If the uploaded content is a duplicate, the framework creates a logical link instead of storing another physical copy.
- When a file is requested, the user submits credentials again and identifies the target file.
- The system retrieves the stored verification value and regenerates a new verification value from the current request.
- Both values are compared.
- If the values match, the encrypted file is released, the convergent encryption key is obtained, and the original data is recovered; otherwise, access is denied.

Another methodological feature of the proposed work is that authentication is performed in two stages. The first stage is conventional user authentication during registration and login. The second stage is file-specific ownership verification during download or retrieval. This layered design is important because it separates general system access from data access authorization. A user may be valid at the application level but still be denied access to a specific file if the ownership verification value does not match. This improves control over deduplicated storage by ensuring that access decisions are based on both identity validation and file-linked proof of legitimacy.

Workflow and Algorithm of the Proposed Model

Figure 2 illustrates the workflow of the proposed model. The process begins with user registration and login, followed by file upload. During upload, the system generates a file token and file tag, which are combined with user credentials to compute a secure verification value. This value is stored in the verification repository and later used during file access. When a user requests a file, the system regenerates the verification value and compares it with the stored value. Access is granted only if both values match; otherwise, the request is rejected. This workflow ensures that deduplicated cloud data is released only after ownership-aware verification.

The proposed model follows a structured workflow to ensure that access to deduplicated cloud data is granted only to a legitimate and authorized user. The workflow combines user authentication, file metadata generation,

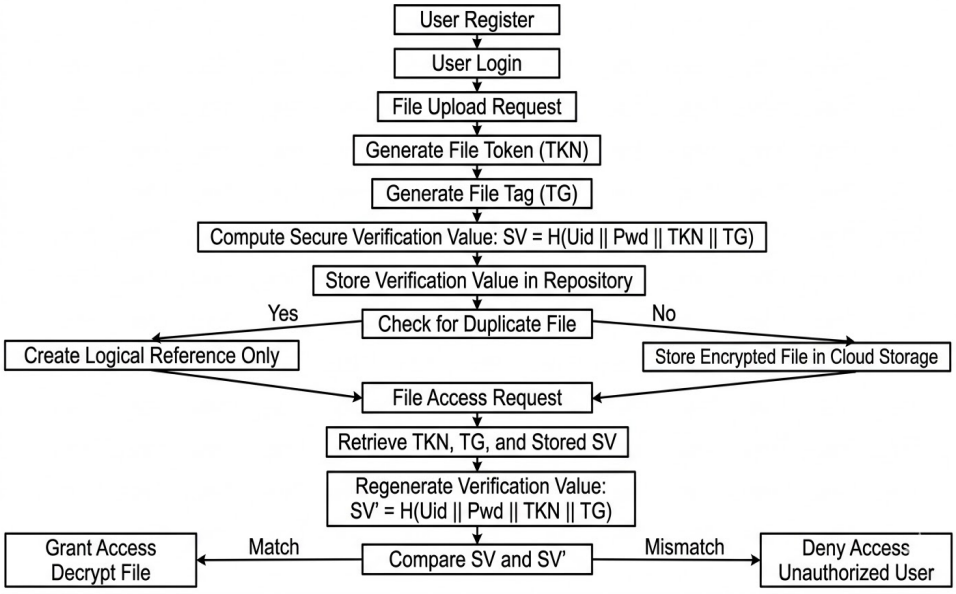


Figure 2: Workflow of the proposed secure ownership verification model for accessing deduplicated cloud data

secure verification value creation, duplicate detection, and ownership-based access validation. Its purpose is to ensure that storage optimization through deduplication does not weaken the security of file retrieval.

Stepwise Workflow of the Proposed Model

The operational flow of the proposed model consists of the following stages:

Stage 1: User Registration and Login

At the beginning, the user registers with the cloud application by providing the required identity details and login credentials. Once registration is completed, the user is allowed to log in to the system. This stage establishes the initial authentication layer and enables the framework to maintain a secure record of valid users.

Stage 2: File Upload and Metadata Generation

After successful login, the user uploads a file to the cloud system. During this process, the framework generates the metadata required for secure deduplication and future access verification. In particular, the system creates a file token and a file tag, which are used to identify and process the uploaded file within the deduplication environment.

Stage 3: Verification Value Creation

Once the file-related metadata is generated, the system computes a secure verification value by combining the user identity, password, token, and tag through a cryptographic hash process. This value becomes the ownership-linked control parameter associated with the uploaded file. It is stored in the verification repository for later use during file access.

Stage 4: Duplicate Detection

The system checks whether the uploaded file already exists in storage. If the file is new, the encrypted version of the file is stored in the cloud. If the file is identified as a duplicate, the system avoids storing another physical copy and instead creates a logical association between the user andw the already stored file. This preserves storage efficiency while maintaining access traceability.

Stage 5: File Access Request

When the user later requests a file, the framework initiates a second verification stage. At this point, the user must provide valid credentials and identify the required file. This request triggers the ownership-verification procedure before the file is released.

Stage 6: Regeneration of Verification Value

The framework retrieves the stored token, tag, and previously saved verification value corresponding to the requested file. Using the credentials submitted during the access request, the system regenerates a new verification value through the same cryptographic process used at upload time.

Stage 7: Ownership Validation

The newly generated verification value is compared with the value stored in the verification repository. If both values are identical, the system confirms that the requester is the legitimate user associated with the file. If the values do not match, the request is treated as invalid and access is denied.

Stage 8: Controlled File Release

After successful ownership validation, the system retrieves

the required decryption information, reconstructs the file for the authorized user, and grants access. Thus, file release is controlled not only by login authentication, but also by ownership-aware verification at the time of retrieval.

Algorithm 1: Secure Ownership Verification for Deduplicated Cloud Data Access

Input:

User ID Uid, Password Pwd, File F

Output:

Access Granted / Access Denied

Steps:

- Register the user in the system and store login credentials.
- *Authenticate the user during login.*
- *Accept file F from the authenticated user.*
- *Generate file token TKN.*
- *Generate file tag TG.*
- *Compute secure verification value:*

$$SV = H\left(Uid \,||\, Pwd \,||\, TKN \,||\, TG\right)$$

- *Store SV in the ownership verification repository.*
- *Check whether file F already exists in the deduplicated cloud storage.*
- *If file F is not present, store the encrypted file in the cloud.*
- *If file F already exists, create only a logical reference for the user.*
- *When the user requests file access, accept Uid, Pwd, and file identifier Fid.*
- *Retrieve the corresponding stored TKN, TG, and stored SV.*
- *Regenerate a new verification value:*
  $SV' = H\left(Uid \,||\, Pwd \,||\, TKN \,||\, TG\right)$
- *Compare SV' with stored SV.*
- *If $SV' = SV$, authenticate ownership and grant file access.*
- *Retrieve the required decryption key and reconstruct the original file.*
- *Else, deny access and terminate the request.*

Mathematical Representation

The workflow of the proposed model can be expressed through two simple phases.

Upload-Time Verification Value Generation

During file upload, the system computes:

$$SV = H\left(Uid \,||\, Pwd \,||\, TKN \,||\, TG\right)$$

where:

- SV = secure verification value
- Uid = user identity
- Pwd = user password
- TKN = token generated for the file
- TG = tag assigned to the file
- H = secure cryptographic hash function

Access-Time Ownership Validation

At the time of file retrieval, the system computes:

$$SV' = H\left(Uid \,||\, Pwd \,||\, TKN \,||\, TG\right)$$

The authorization condition is:

$$\text{If } SV = SV', \text{ then access is granted;}$$

$$\text{otherwise, access is denied.}$$

This mathematical condition ensures that file access depends on the correct combination of user credentials and file-specific metadata.

The workflow of the proposed model is significant because it introduces a two-level protection mechanism. First, the user must be authenticated at the system level. Second, the user must pass a file-specific ownership check before deduplicated data is released. This layered structure is especially important in deduplicated cloud storage, where multiple users may reference the same stored object. Without such a verification process, a weak ownership model could allow unauthorized retrieval of confidential data.

The proposed workflow therefore strengthens the security of deduplicated storage in three ways. First, it binds user identity directly with file metadata. Second, it prevents access based on incomplete or unrelated ownership claims. Third, it preserves the storage advantages of deduplication without compromising access control. As a result, the workflow provides a practical and security-oriented foundation for controlled data access in modern cloud environments.

Results and Discussion

The performance of the proposed model was evaluated in terms of computational time during file access and download, using different file sizes ranging from 1 KB to 5 KB. The implementation was carried out in a cloud-based environment using C#.NET, and the observed results were compared with those of CAE (Wang et a., 2025) and Dekey (Bronson, F. A. X., et al., 2023). The objective of this evaluation was to determine whether the proposed ownership-aware access verification model could improve file access security without introducing excessive overhead during retrieval.

Table 1 shows that the computational time increases for all methods as the file size grows, which is expected because larger files require greater processing effort during retrieval. However, the proposed model consistently records the lowest computational time at every tested file size. At 1 KB, the proposed model requires 2ms, compared with 5 ms for CAE and 8 ms for Dekey. The same pattern continues at other size of data, indicating that the proposed verification

Table 1: Computational time of the proposed model and existing techniques

File size	*CAE (ms)*	*Dekey (ms)*	*Proposed Model (ms)*
1 KB	5	8	2
2 KB	9	15	4
3 KB	14	23	5
4 KB	18	30	7
5 KB	23	38	9

Table 2: Performance improvement of the proposed model over existing techniques

File size	*Improvement over CAE*	*Improvement over Dekey*
1 KB	10.40%	26.80%
2 KB	5.00%	23.49%
3 KB	3.39%	14.71%
4 KB	2.86%	13.53%
5 KB	2.11%	11.71%

mechanism remains computationally efficient even as the file size increases.

The results suggest that the additional ownership-aware validation introduced by the proposed model does not create a heavy retrieval burden. Instead, the framework maintains lower access time while also enforcing a stronger file-specific verification process. This is an important finding because it demonstrates that improved access security in deduplicated cloud storage does not necessarily come at the cost of reduced efficiency. Among the compared methods, Dekey shows the highest computational time throughout the experiment, while CAE performs closer to the proposed model but remain slower at all tested file sizes.

Table 2 provides the percentage reduction in computational time achieved by the proposed model over the baseline approaches. The highest improvement is observed against Dekey, where the proposed model reduces computational time by 26.80% at 1 KB and still maintains an 11.71% advantage at 5 KB. When compared with CAE, the percentage improvement is smaller but remains consistent across all file sizes. These findings show that the proposed model preserves a stable performance advantage across the full experimental range.

Another important observation is that the percentage improvement becomes smaller as the file size increases. This does not weaken the value of the proposed method; rather, it indicates that all methods experience higher processing cost at larger volumes of data. Even under those conditions, the proposed model still remains the fastest among all compared approaches. This stability is important for practical cloud deployment, where file sizes may vary significantly and access-control mechanisms must remain efficient across different workloads.

Security Analysis and Discussion

While the experimental measurements demonstrate computational efficiency, the importance of the proposed model also lies in its security strength. The security aspect is discussed here as a comparative analytical assessment based on the design characteristics of the method. The proposed model generates a verification value using the combination of user ID, password, token, and file tag, and access is granted only when the newly generated value matches the stored record. This makes the access decision dependent on both user-specific credentials and file-specific metadata, thereby strengthening ownership-aware authorization.

Table 3 indicates that the proposed model offers stronger ownership-aware access protection than the compared approaches. The key reason is that the method does not rely solely on a general proof of file possession. Instead, it binds the user's identity and credentials with the token and tag associated with the file, thereby creating a stronger verification relationship. As a result, the model provides better protection against false ownership claims, unauthorized duplicate access, and weak retrieval-stage authorization. This is particularly important in deduplicated cloud environments, where multiple users may point to the same stored object and the system must distinguish legitimate ownership from unauthorized access attempts.

A major security advantage of the proposed design is that verification is performed during file retrieval, not merely at upload time. This means that even after successful login,

Table 3: Comparative security analysis of the proposed model

Security parameter	*CAE*	*Dekey*	*Proposed model*
Resistance to false ownership claim	Moderate	Moderate	High
Credential-linked access validation	No	No	Yes
File-specific verification before access	Limited	Limited	Yes
Protection against unauthorized duplicate access	Moderate	Moderate	High
Dependence on single file-derived evidence	High	Moderate	Low
Binding of user identity with file metadata	No	Partial	Strong
Access control at retrieval stage	Limited	Limited	Strong
Suitability for ownership-aware authorization	Moderate	Moderate	High

the user must still pass a second file-specific validation check before the system releases the requested data. Such a layered design improves practical access control because it ensures that system-level authentication alone is not sufficient to obtain a deduplicated file. Instead, the user must also prove a valid ownership-linked association with that specific file.

The computational and security analyses show that the proposed model performs well in two critical aspects of deduplicated cloud storage: efficiency and secure access enforcement. The experimental results confirm that the model maintains lower computational time than the compared approaches across all tested file sizes. At the same time, the security analysis indicates that the method provides stronger ownership-aware access protection because it links retrieval authorization to both user credentials and file metadata. This combination is important in modern cloud information security, where storage optimization must not weaken confidentiality or allow unauthorized retrieval of shared cloud data. Therefore, the results indicate that the proposed model not only improves computational efficiency during deduplicated file access, but also strengthens ownership-aware security by enforcing a more reliable verification process at the point of retrieval.

Conclusion

The overall findings of this research work confirm that the proposed ownership-aware verification model provides an effective balance between security and performance in deduplicated cloud storage environments. The experimental results show that the model consistently achieves lower computational time than CAE and Dekey across different file sizes, demonstrating that stronger access verification does not necessarily introduce excessive retrieval overhead. At the same time, the design improves security by linking user credentials, token information, and file-specific metadata to the access decision, thereby strengthening protection against false ownership claims and unauthorized file retrieval. Unlike approaches that rely mainly on general proof-of-possession logic, the proposed method enforces verification directly at the retrieval stage, which makes access control more practical and ownership-aware. This added verification layer is especially valuable in deduplicated storage settings, where multiple users may logically reference the same physical file. The analytical security assessment further supports that the model offers stronger user–file binding, better retrieval-stage authorization, and more reliable ownership enforcement than the compared methods. Thus, the proposed approach not only improves computational efficiency but also enhances secure access management in cloud deduplication systems, making it a practical solution for modern cloud data protection.

Acknowledgement

I sincerely acknowledge our principal and secretary for providing the facility to complete this paper Successfully.

References

Bronson, F. A. X., Raja, V. S. S., Elumalai, S., Velan, G. S., & Solainayagi, P. (2023). Integrated storage allocation and privacy preserve deduplication process over cloud using Dekey method. In A. K. Rana, A. Rana, S. Dhawan, S. Sharma, & A. A. Elngar (Eds.), Convergence of deep learning and artificial intelligence in internet of things (pp. 39–64). CRC Press. https://doi.org/10.1201/9781003355960-3

Dave, J., Ram, K. R., Patil, P., Patil, H., Borole, S., & Panda, S. (2025). Secure and efficient ownership verification for deduplicated cloud computing systems. *Journal of Cloud Computing, 14*, 20. https://doi.org/10.1186/s13677-025-00743-y

Dave, J., Ram, K. R., Patil, P., Patil, H., Borole, S., Chinni, V. K., & Patil, S. (2026). Blind and bidirectional ownership verification for deduplicated cloud computing systems. *Future Generation Computer Systems, 175*, 108082. https://doi.org/10.1016/j.future.2025.108082

Jin, C., Xu, Y., Qin, W., Zhao, J., Kan, G., & Zeng, F. (2024). A blockchain-based auditable deduplication scheme for multi-cloud storage. *Peer-to-Peer Networking and Applications, 17*(5), 2870–2883. https://doi.org/10.1007/s12083-024-01734-7

Lee, M., & Seo, M. (2023). Secure and efficient deduplication for cloud storage with dynamic ownership management. Applied Sciences, 13(24), 13270. https://doi.org/10.3390/app132413270

Prajapati, P., Patel, M., & Jinwala, D. (2022). A review on secure data deduplication: Cloud storage security issue. Journal of King Saud University – Computer and Information Sciences, 34(8), 5300–5314.

Tang, X., Guo, C., Choo, K.-K. R., Jiang, X., & Liu, Y. (2024). A secure and lightweight cloud data deduplication scheme with efficient access control and key management. *Computer Communications, 222*, 209–219. https://doi.org/10.1016/j.comcom.2024.05.003

Uma, G. (2018). Survey on Data Deduplication Techniques used for Efficient Management of Cloud Storage. International Journal Computer Science Management Studies, 7(3).

Uma, G., & Jayasimman, L. (2018). Enhanced convergent encryption key generation for secured data deduplication in cloud storage. In Journal of Physics: Conference Series (Vol. 1142, No. 1, p. 012006). IOP Publishing.

Vijayakumar, D., Srinivasagan, K. G., & Vivekrabinson, K. (2025). Enhancing cloud storage security through blockchain-enabled data deduplication and auditing with a fair payment. *Peer-to-Peer Networking and Applications, 18*, 147. https://doi.org/10.1007/s12083-025-01970-5

Wang, C., Zhang, G., Qi, H., & Chen, B. (2025). A CAE model-based secure deduplication method. Scientific reports, 15(1), 24605. https://doi.org/10.1038/s41598-025-09788-0

Wang, M., Xu, L., Hao, R., & Yang, M. (2023). Secure auditing and deduplication with efficient ownership management for cloud storage. *Journal of Systems Architecture, 142*, 102953. https://doi.org/10.1016/j.sysarc.2023.102953

Yang, Z., Tian, W., Li, R., Ye, X., & Xu, Z. (2024). Who owns the cloud data? Exploring a non-interactive way for secure proof of

ownership. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (pp. 1494–1499). IEEE. https://doi.org/10.1109/TrustCom63139.2024.00206

Yu, J., Shen, W., & Zhang, X. (2024). Cloud storage auditing and data sharing with data deduplication and private information protection for cloud-based EMR. *Computers & Security, 144*, 103932. https://doi.org/10.1016/j.cose.2024.103932

Zhou, T., Wang, L., Liang, M. (2026). ZKNiS-PoW: A privacy-preserving proof of ownership scheme for secure cloud storage. *Array, 29*, 100627. https://doi.org/10.1016/j.array.2025.100627

Zhou, T., Wang, L., Liang, M., & Li, M. (2025). MiS-PoW: Mirror-Selected Non-Interactive Proof of Ownership for Cloud Storage. *Applied Sciences, 15*(22), 11897. https://doi.org/10.3390/app152211897