iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82000

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Secure Cloud Storage with Encryption and Data Integrity Mechanisms

Sheela Rani C M[1], Naveen K[2], Likhith Kumar R P[3], Mahamad Usen[4], Akash B V[5]
[1]*Assistant Professor, Dept. of CSE Sapthagiri College of Engineering*
[2, 3, 4, 5]*Dept. of CSE Sapthagiri College of Engineering*

Abstract: ***Remote cloud infrastructure for sensitive data storage continues to pose unresolved technical and governance challenges as cloud adoption grows across healthcare, finance, and public administration. This survey examines ten recent research contributions from 2021 to 2025 that address encryption algorithms, cryptographic key lifecycle management, access control enforcement, and integrity verification mechanisms in cloud file storage environments. The examined works encompass symmetric block ciphers, public-key exchange protocols, hybrid combinations, client-controlled encryption architectures, hardware-assisted key protection, and distributed fault-tolerant storage models. Comparative analysis consistently reveals that the gap between theoretically sound cryptographic designs and their practical deployment usability remains the central unsolved challenge.***

End-user key management burdens, computational overhead from asymmetric operations, and the absence of lightweight implementations for resource-constrained environments each contribute to this divide. The findings inform the design of a PHP and MySQL-based secure cloud file storage prototype that performs AES-256 encryption on the client side before any data reaches the server.

Keywords: cloud storage security, AES-256, RSA, hybrid cryptography, client-side encryption, key management, access control, data integrity, homomorphic encryption, cloud computing

I. INTRODUCTION

The large-scale migration of institutional and personal data onto third-party cloud platforms has accelerated over the past decade, driven by reductions in capital expenditure, improvements in collaborative workflows, and the proliferation of mobile devices demanding anywhere-accessible storage. Major cloud service providers now hold petabytes of sensitive records on behalf of governments, healthcare organisations, financial institutions, and private individuals.

The inherent convenience of this arrangement carries a structural trade-off: once data crosses from a user's device into a provider's infrastructure, direct oversight of how data is stored, who may inspect it, and under what conditions it may be disclosed is effectively relinquished.

The predominant encryption model adopted by commercial providers relies on server-side key management: the provider generates and holds the cryptographic keys protecting stored files, decrypts data when delivering it to authenticated users, and assumes full responsibility for key store security. While operationally convenient, this model introduces a well-documented vulnerability. Any compromise of the provider's key management infrastructure through external intrusion, misconfigured access policies, or a malicious insider grants an adversary immediate access to the plaintext content of all managed files. Client-side encryption addresses the structural weakness of provider-managed keys by performing all cryptographic operations on the user's device prior to data transmission.

Under this model the provider receives only ciphertext, and even a complete compromise of provider infrastructure yields no readable content. The trade-off, however, is that the user must independently manage encryption keys, introducing usability burdens and recovery risks that have proven to be substantial barriers to real-world adoption. This survey was conducted to contextualise the design of a secure cloud file storage prototype developed as a major engineering project. Ten peer-reviewed publications from 2021 to 2025 were selected on the basis that they directly address the technical questions the prototype must resolve: which encryption algorithm delivers the best balance of security and processing efficiency, how keys should be derived and stored, how access control should be enforced without relying on server-side plaintext inspection, and how data integrity can be verified without transmitting unencrypted content.

II. LITERATURE SURVEY

A. Rajan, Mittal, and Gupta (2025)

This study investigates a cloud file storage architecture in which all cryptographic operations are carried out within the browser before any payload reaches the remote storage backend.

The implementation employs the AES symmetric cipher through a widely adopted JavaScript cryptographic library, with a user-supplied password forming the basis for key derivation.

Files are segmented before encryption to limit peak memory consumption, and multi-factor authentication is enforced to guard against credential theft. The study presents a practical argument for relocating the encryption boundary to the client side, but a one-megabyte ceiling on uploaded files and an eight-character maximum password length are imposed by the prototype. No mechanism assists users who lose their passphrase in recovering stored data, and no pathway exists for multiple users to share access to a single encrypted file.

B. Brundashree et al. (2025)

Brundashree and co-authors present a layered encryption design in which the file payload is secured with a freshly generated AES-256 session key, and that session key is subsequently protected by encapsulation within an RSA public-key operation before storage. This separation allows the computationally efficient symmetric cipher to process bulk data while the asymmetric algorithm handles only the smaller task of securing the session key.

The stored data is additionally fragmented across several cloud storage nodes using a sharding approach, but distributing fragments across multiple nodes introduces storage costs that scale faster than the underlying data volume, and procedures for revoking user access or rotating asymmetric keys are addressed only at a high level.

C. Myers et al. (2024)

Myers and colleagues conduct a systematic side-by-side evaluation of four algorithms: the AES block cipher, the RSA asymmetric scheme, the Blowfish block cipher, and the bcrypt password-hashing function. The evaluation platform is a simulated cloud environment built on container orchestration software with workloads varying from a small number of sessions to one thousand users.

AES achieves the lowest latency across all load levels and maintains stable processor utilisation. RSA incurs latency an order of magnitude higher than AES under moderate load. Bcrypt is deliberately designed to be slow; while useful for credential storage, this throughput characteristic is incompatible with high-volume file encryption. The authors recommend combining AES for data encryption, bcrypt for password hashing, and RSA for key transport.

D. Ravindrakumar (2024)

This work surveys the principal families of encryption applicable to cloud storage and proposes the Encryption and Cryptographic Resource Management (ECRM) framework.

The framework organises users into groups under a designated administrator who controls membership, issues cryptographic credentials, and manages revocation.

Access to stored files is gated by digital signature verification, and an auditing database maintains the authoritative record of valid credentials. The principal limitation is that pairing-based cryptography, while mathematically expressive, is considerably more computationally intensive than standard AES and lacks the breadth of library and hardware acceleration support available for conventional symmetric ciphers.

E. Zahir (2025)

Zahir approaches cloud storage security from a risk analysis perspective, combining a review of academic publications with examination of several widely-reported breaches and outages. The analysis is structured around the three classical security objectives of confidentiality, integrity, and availability. A 2023 vulnerability in a widely used file transfer application was exploited across dozens of organisations through a third-party vendor. The author uses this case to argue that the shared responsibility model breaks down when customers cannot independently audit the security posture of every vendor in their supply chain. User-facing key management is identified as the central unsolved challenge.

F. Kim et al. (2025)

This contribution addresses a scenario in which data must be protected against the possibility that any individual cloud provider might be compromised or act contrary to the data owner's interests.

The approach fragments each encrypted file into shares distributed across providers from different jurisdictions, applying a threshold scheme in which any subset meeting a defined minimum size is sufficient to reconstruct the original, while no smaller subset yields information about the underlying plaintext.

The principal disadvantage is operational complexity: distributing and retrieving shares across multiple providers incurs network overhead growing with the number of participating nodes.

G. Singh et al. (2023)

Singh and colleagues examine an enterprise scenario in which file access rights depend not on static role assignments but on a combination of contextual factors that may evolve over time.

The proposed framework pairs AES for encrypting file content with RSA for securing session keys, and adds an attribute-based encryption layer allowing access policies to incorporate factors such as the user's current location, the security classification of the device in use, the time of day, and the sensitivity level of the requested file. Performance analysis reveals that overhead from evaluating attribute policies at access time is modest when the number of attributes and policy rules is limited, but grows measurably as policy complexity increases.

H. Kaur, Kaur, and Verma (2022)

This paper examines the challenge that strong client-side encryption, while making content opaque to the storage server, simultaneously prevents the server from enforcing access rules in the conventional manner.

The study reviews three technical approaches: proxy re-encryption, which allows the server to transform ciphertext without accessing the plaintext; attribute-based encryption, which embeds the access policy within the ciphertext itself; and searchable encryption, which permits keyword retrieval without revealing the keyword or content to the server. Each approach carries distinct performance characteristics.

The survey concludes that no single technique is universally superior and the best choice depends on the specific access control requirements of the deployment.

I. Chen, Zhang, and Zhu (2021)

Chen and co-authors investigate whether network coding techniques can serve dual roles as a storage efficiency mechanism and an integrity assurance layer for cloud-stored encrypted data.

In the proposed scheme, an encrypted file is transformed into coded fragments using a linear network code and distributed across multiple servers.

Any subset of fragments exceeding a defined recovery threshold suffices to reconstruct the original, enabling the system to tolerate a certain number of server failures or data corruption events without requiring full redundant copies. The practical limitation is that network coding is a specialised technique unsupported by standard cloud storage APIs, requiring custom middleware at both upload and retrieval stages.

J. Crocker and Querido (2021)

The final paper addresses the problem of key custody from a hardware perspective. The central argument is that software-based key management is inherently limited because any key that exists as data in a general-purpose computing environment is potentially exposed to an adversary who gains sufficient system access. Hardware Security Modules address this by providing tamper-resistant physical devices within which keys are generated, stored, and used, but from which key material itself never exits in plaintext. The proposed two-factor authentication architecture combines a knowledge credential with a physical possession credential and routes all cryptographic operations through the HSM. Acknowledged limitations include hardware procurement cost and reduced accessibility for users who cannot carry hardware tokens.

III. COMPARATIVE SUMMARY OF REVIEWED WORKS

Table 1 below consolidates the ten reviewed papers across four dimensions: authors and publication year, the principal technical methodology employed, and the most significant gap or limitation identified.

No.	Paper Title	Authors & Year	Core Methodology	Principal Gap / Limitation
1	Secured Cloud Storage with Client-Side Encryption	Rajan et al., 2025	AES via CryptoJS; per-file password derivation; chunk processing; MFA; Cloudinary backend	File cap of 1 MB; 8-char password limit; no automated key recovery or multi-user support
2	Secure File Storage on Cloud Using Hybrid Cryptography	Brundashree et al., 2025	AES-256 for payload; RSA for key wrapping; RBAC; data sharding across AWS S3; MFA	RSA overhead in key exchange; storage bloat from sharding; revocation only partially handled
3	Evaluation of Encryption Algorithms in Cloud-Based Password Storage	Myers et al., 2024	Benchmarked AES, RSA, Blowfish, bcrypt on Kubernetes (10–1000 users); ANOVA analysis	No single algorithm meets all criteria; bcrypt latency unacceptable at enterprise scale
4	Data Encryption Techniques for Securing Cloud Storage and Communication	Ravindrakumar, 2024	ECRM framework; JPBC-based group-signature access control; upload/download/encrypt metrics	Pairing-based crypto impractical at scale; key distribution unresolved; GDPR alignment thin
5	Cloud Storage Security: Risks and Solutions	Zahir, 2025	Systematic review + MOVEit/Dropbox incident analysis; CIA triad mapping; SSE and PoR review	Supply-chain exposure unaddressed; user key-management burden remains the core open problem
6	Distributed Encryption for Multi-Cloud Storage	Kim et al., 2025	Threshold cryptography; secret sharing; Byzantine fault tolerance; cross-provider key distribution	Significant network overhead; management of distributed keys across jurisdictions not fully defined
7	Hybrid Encryption and Access Control for Enterprise Cloud	Singh et al., 2023	AES + RSA + ABE; contextual ABAC (time, location, device); attribute lifecycle benchmarking	Latency rises with attribute count; delegation scope limits and usability not deeply explored
8	Access Control and Permission Management in Encrypted Cloud Storage	Kaur et al., 2022	Proxy re-encryption; ABE; searchable encryption with integrated access control; revocation study	Re-encryption overhead significant; delegation breadth constraints insufficiently analysed
9	Secure Cloud Storage Using Network Coding and Cryptographic Verification	Chen et al., 2021	Network-coded encrypted fragments; formal security proofs; fault-tolerant recovery protocol	Specialised coding expertise required; segment synchronisation costs not fully quantified
10	Hardware-Based Security and MFA in Cloud Storage	Crocker & Querido, 2021	HSM key custody; two-factor auth (password + token); OAuth 2.0 token exchange; latency testing	Hardware provisioning cost; accessibility barriers for diverse user groups not examined

Table 1: Comparative summary of reviewed research papers

IV. DISCUSSION

A. *AES-256 as the Practical Encryption Standard*

Every paper in this review that involves symmetric file encryption selects AES, and the majority favour the 256-bit key length. The benchmark data reported by Myers et al. provides the most rigorous quantitative support: under conditions simulating enterprise-level concurrency, AES delivers sub-five-millisecond round-trip times and stable processor utilisation. The consistency of this finding across independently conducted studies from different institutions with different evaluation methodologies supports treating AES-256 as the default choice for encrypting file payloads in cloud storage contexts, at least until quantum-resistant algorithms mature to practical deployment readiness.

B. Asymmetric Algorithms as Key Transport Mechanisms

RSA appears in several reviewed architectures, but never as the primary encryption engine for file content. Its role in every case is to protect the symmetric session key during transmission or storage, exploiting the mathematical property that the public component of an RSA key pair can be freely distributed while the private component remains exclusively with the intended recipient. This separation resolves the classic key distribution problem in symmetric cryptography without imposing RSA's latency penalty on bulk data operations. A well-designed cloud storage system treats the two algorithm families as complementary, deploying each where its properties are advantageous.

C. Key Management as the Central Unsolved Problem

Across the ten reviewed papers, the challenge of managing cryptographic keys in a manner that is both secure and practically sustainable emerges as the most consistently identified gap. Papers approach the problem from different angles and each arrives at the same conclusion: existing solutions either impose unacceptable complexity on users or concentrate key custody in a location that becomes a single point of failure. The HSM-based approach of Crocker and Querido relocates the trust boundary to hardware but does not eliminate it, and introduces procurement and operational barriers that limit applicability. No reviewed paper presents a key management architecture simultaneously secure, accessible without specialist knowledge, scalable to large user populations, and recoverable when credentials are lost.

D. The Server-Side Decryption Vulnerability

The structural limitation of provider-managed key custody is articulated most clearly by Zahir and Rajan et al., though it is implicitly recognised by every paper proposing client-side encryption as an alternative. When a provider holds the keys and performs decryption on behalf of users, the provider's infrastructure becomes a target whose compromise simultaneously yields access to all stored content. The 2023 file transfer application breach illustrates how this vulnerability can be exploited by targeting a third-party vendor with access to the provider's systems. Distributing file fragments across multiple providers reduces but does not eliminate this risk, since the attacker's objective shifts to compromising the threshold number required for reconstruction.

E. Performance and Usability Trade-Offs

The reviewed literature converges on a familiar trade-off: measures that strengthen security typically impose performance costs or usability penalties. Bcrypt's intentional computational slowness defeats brute-force attacks but renders it unsuitable for encrypting frequently accessed files. Attribute-based encryption enforces fine-grained access policies but introduces latency that grows with policy complexity. Homomorphic encryption, discussed by Ravindrakumar as a future direction, would permit computations on ciphertext without decryption but currently incurs overhead several orders of magnitude greater than standard encryption. Algorithm selection must therefore be guided by the specific threat model and usage pattern of the target deployment.

F. Absence of Lightweight Deployable Implementations

A consistent observation across the reviewed body of work is that proposed systems are designed for enterprise or large-scale cloud environments: distributed storage across multiple regions, container orchestration platforms, Hardware Security Module infrastructure, or specialised cryptographic middleware. None of the reviewed papers proposes and evaluates an architecture suited to deployment on a standard shared web hosting environment using conventional scripting languages and a relational database. This gap is significant because a substantial proportion of institutional data is managed by small organisations that lack resources for enterprise cloud infrastructure but still need to protect sensitive files. The present project addresses this gap by targeting a PHP and MySQL implementation deployable on commodity hosting infrastructure.

V. CONCLUSION

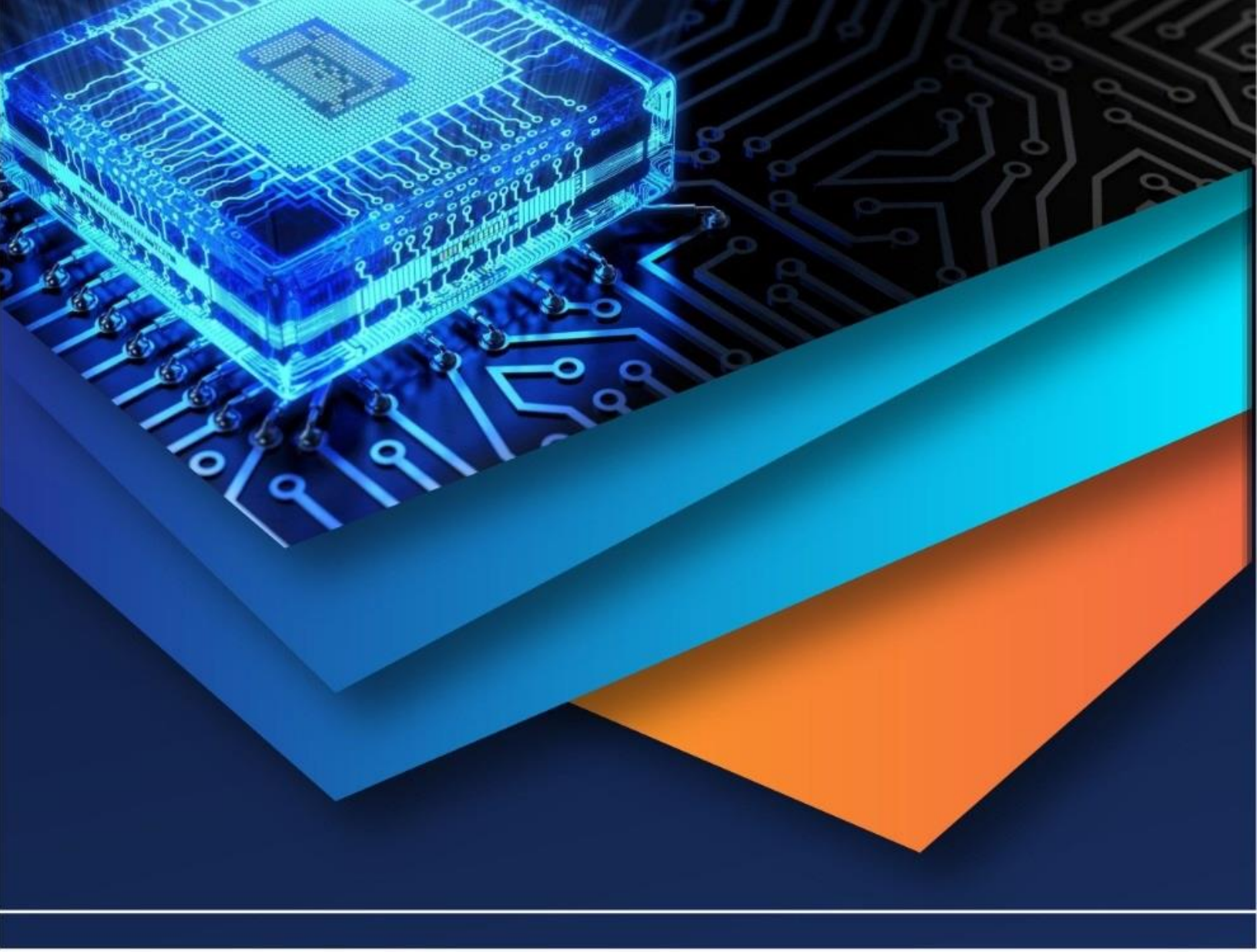
This survey has examined ten research contributions addressing cryptographic protection for cloud-stored files, spanning 2021 to 2025 and covering algorithm benchmarking, hybrid encryption architectures, client-side key management, access control enforcement, hardware-assisted key custody, and fault-tolerant distributed storage. AES-256 is the most practical symmetric cipher for encrypting file payloads, offering a combination of security margin and processing efficiency that no alternative examined in the reviewed literature surpasses for this purpose. Asymmetric algorithms such as RSA are best applied to key transport rather than bulk encryption. Hybrid architectures combining both families address the symmetric key distribution problem while keeping computational costs manageable.

The central unsolved challenge identified across the reviewed body of work is key management: how to store, distribute, rotate, and recover cryptographic keys in a manner simultaneously secure, accessible to non-expert users, and resilient to common failure modes such as forgotten passwords or lost devices. Every reviewed paper acknowledges this problem; none presents a fully satisfying resolution. The hardware security module approach of Crocker and Querido comes closest to a principled solution but at a cost in procurement complexity and accessibility that limits its general applicability.

A further gap is the absence of lightweight implementations suited to small-scale institutional deployments. The present project contributes to filling this gap by combining AES-256 client-side encryption with session-based access control in a PHP and MySQL architecture deployable on standard hosting infrastructure. Future extensions could incorporate post-quantum cryptographic primitives, integrate machine-learning-based anomaly detection, and adopt blockchain-based audit logging to provide tamper-evident records of all file operations.

REFERENCES

[1] G. V. Rajan, R. Mittal, and A. Gupta, "Secured cloud storage with client side encryption," Galgotias University, Greater Noida, India, 2025. Available: https://ssrn.com/abstract=5870642
[2] A. Brundashree et al., "Secure file storage on cloud using hybrid cryptography," IJSAT, vol. 16, no. 2, pp. 1–11, 2025.
[3] J. T. Myers et al., "Evaluation of encryption algorithms in cloud-based password storage systems," Journal of Cloud Security and Privacy, vol. 14, no. 3, pp. 187–205, 2024.
[4] Ravindrakumar, "Data encryption techniques for securing cloud storage and communication," ShodhKosh, vol. 5, no. 4, pp. 861–869, 2024.
[5] R. M. Zahir, "Cloud storage security: risks and solutions," Preprints.org, Nov. 2025.
[6] C. Kim, J. Park, Y. Lee, and S. Wang, "Distributed encryption mechanisms for multi-cloud storage environments," IEEE Trans. Cloud Comput., vol. 13, no. 2, pp. 156–170, 2025.
[7] A. Singh et al., "Hybrid encryption and access control frameworks for enterprise cloud storage," IJISP, vol. 17, no. 3, pp. 234–252, 2023.
[8] H. Kaur, J. Kaur, and A. Verma, "Access control mechanisms and permission management in encrypted cloud storage," IEEE Access, vol. 10, pp. 45678–45695, 2022.
[9] H. Chen, Y. Zhang, and B. Zhu, "Secure cloud storage using network coding and cryptographic verification," IEEE Trans. Cloud Comput., vol. 9, no. 4, pp. 1045–1060, 2021.
[10] B. Crocker and S. Querido, "Hardware-based security and multi-factor authentication in cloud storage systems," Journal of Cybersecurity and Privacy, vol. 11, no. 2, pp. 89–107, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY