Secure Data Sharing in Cloud Computing: A Cryptographic Review and Methodology

Bilal ABDULHADI
Computer Engineering Department
Biruni University
251132903@st.biruni.edu.tr

***Abstract*—Cloud computing has changed modern data management by providing scalable storage, on-demand services, and flexible collaboration across organizations. However, when sensitive data is stored and shared through third-party cloud service providers, the data owner loses direct physical control over the infrastructure. This creates important security concerns related to confidentiality, integrity, authentication, access control, and provider trust. This paper presents a technical review of secure data sharing in cloud computing and proposes a practical cryptographic methodology for protecting shared cloud data. The study discusses the main threat model of cloud storage, including the honest-but-curious provider assumption, unauthorized access, insider threats, and data tampering. It then analyzes major security techniques such as Advanced Encryption Standard (AES), RSA-based key exchange, SHA-256 hashing, digital signatures, and audit logging. The proposed approach combines client-side encryption, secure key sharing, integrity verification, and access auditing to reduce trust in the cloud provider and improve accountability. The paper concludes that secure cloud sharing requires more than storing encrypted files; it requires a balanced framework where privacy, usability, revocation, and system performance are considered together.**

***Index Terms*—Cloud Computing, Secure Data Sharing, Cryptography, Access Control, Data Integrity, AES, RSA, SHA-256, Cloud Security.**

I. Introduction

Cloud computing has become a central model for modern information technology because it allows users and organizations to access configurable computing resources such as storage, servers, networks, and applications without owning the underlying infrastructure. This model improves scalability, reduces operational cost, and supports collaboration among distributed users [1]. In academic, enterprise, and governmental environments, cloud platforms are now commonly used not only for storage but also for active data sharing.

Despite these advantages, cloud computing introduces a major security challenge: data owners often store sensitive information on infrastructure operated by a third-party cloud service provider (CSP). In traditional local storage, the owner maintains direct control over the physical system. In cloud environments, this control is transferred to external servers, which creates a trust boundary between the data owner and the provider [6].

Secure data sharing is especially difficult because multiple users may need different levels of access to the same data. A researcher may need read-only access to a dataset, a project member may need permission to update a file, and an administrator may need the ability to manage access rights. If these permissions are not controlled properly, shared cloud storage can expose data to unauthorized users, malicious insiders, or external attackers.

The goal of this paper is to review the core security requirements of cloud-based data sharing and present a practical methodology based on cryptographic protection. The proposed methodology focuses on client-side encryption, secure key exchange, integrity checking, and access auditing. These mechanisms are selected because they reduce the need to fully trust the CSP while still allowing authorized users to collaborate efficiently.

II. Background and Related Work

Cloud security research has evolved alongside the growth of cloud services. Early cloud security studies focused mainly on storage availability and data loss prevention, while later work emphasized privacy, integrity verification, access control, and secure collaboration. Armbrust *et al.* described cloud computing as a new computing paradigm that provides elasticity and service availability, but they also identified security and privacy as major obstacles to adoption [1].

Kamara and Lauter introduced the idea of cryptographic cloud storage, where data is protected before it is outsourced to the cloud [2]. Their work emphasizes that encryption should be treated as a foundation for secure cloud systems rather than an optional additional layer. Similarly, Zissis and Lekkas discussed how traditional perimeter-based defenses are not sufficient in cloud environments because cloud systems depend on shared infrastructure and remote administration [6].

A major line of related work focuses on data integrity verification. Shacham and Waters proposed compact proofs of retrievability, allowing a user to verify that cloud-stored data remains intact without downloading the entire file [3]. Wang *et al.* extended this idea through privacy-preserving public auditing, where a third-party auditor can verify cloud data integrity without learning the data content [4].

Another important direction is secure group sharing. Liu *et al.* proposed Mona, a secure multi-owner data sharing scheme for dynamic groups in the cloud [5]. Their work highlights the complexity of managing users who join and leave a group, especially when encryption keys must be updated after access revocation.

This paper builds on these ideas by presenting a clear course-level methodology that combines encryption, hashing, digital signatures, and auditing into a practical secure sharing workflow.

III. Problem Definition

The central problem in cloud data sharing is that sensitive data must be made available to authorized users while remaining protected from unauthorized users, malicious insiders, and even the CSP itself. This problem is not limited to storage security; it also includes secure transmission, permission management, key distribution, and integrity verification.

A. Third-Party Trust Issues

A common assumption in cloud security is the honest-but-curious provider model. Under this model, the CSP is expected to follow the storage and service protocols correctly, but it may still attempt to analyze stored data or metadata for business, surveillance, or operational purposes. This makes plaintext storage risky, even when the provider is technically reliable.

B. Unauthorized Access

Because cloud systems are reachable over the public internet, they have a larger attack surface than isolated local systems. Unauthorized access may come from external attackers exploiting vulnerabilities, malicious insiders with administrative privileges, or legitimate users who obtain permissions beyond their intended role. In shared environments, poor access control can quickly become a serious security failure.

C. Data Integrity Violations

Data integrity means that data remains accurate, complete, and unmodified except by authorized operations. Integrity violations may occur because of hardware errors, software bugs, synchronization failures, or deliberate tampering. In scientific, legal, financial, and medical contexts, even a small unauthorized change may make data unreliable or invalid.

IV. Security Requirements

A secure cloud data-sharing system should satisfy several requirements. These requirements are usually connected to the confidentiality, integrity, and availability model, but sharing scenarios also require strong authentication, authorization, revocation, and accountability.

A. Confidentiality

Confidentiality ensures that data is visible only to authorized parties. In cloud sharing, confidentiality should remain valid even if the CSP stores the file or manages the storage infrastructure. Client-side encryption is one of the strongest ways to achieve this goal because the data is encrypted before it leaves the owner's device [2].

TABLE I
Security Requirements and Suitable Protection Techniques

Requirement	Common Technique
Confidentiality	Client-side encryption using symmetric cryptography such as AES.
Integrity	Cryptographic hashing and proof-based verification.
Authentication	Public-key cryptography and digital signatures.
Access control	Role-based or attribute-based permission management.
Accountability	Audit logs and third-party verification mechanisms.
Revocation	Key rotation, re-encryption, and policy updates.

B. Integrity

Integrity ensures that data has not been changed without authorization. Hash functions and integrity verification protocols are commonly used to detect tampering. In cloud systems, integrity is important because the data owner may not directly observe storage operations or physical disk behavior [3].

C. Authentication and Non-Repudiation

Authentication verifies the identity of the user or entity interacting with the data. Non-repudiation ensures that a user cannot later deny an action they performed. Digital signatures support both properties by linking a message or file to the private key of the signer [8].

D. Access Control and Revocation

Access control defines what each user is allowed to do. In shared cloud storage, common roles include owner, reader, writer, and auditor. Revocation is the process of removing access from a user. This is challenging in encrypted systems because a user who previously received a decryption key may keep a copy of it after being removed from the group [5].

V. Cryptographic and Security Techniques

Cryptographic tools provide the technical foundation for secure cloud data sharing. This section reviews the main techniques used in the proposed methodology.

A. AES for Confidentiality

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm standardized by NIST [7]. Because the same key is used for encryption and decryption, AES is efficient and suitable for encrypting the actual file contents. In secure cloud sharing, AES is commonly used for data-at-rest protection because it offers strong security with practical performance.

B. RSA for Secure Key Exchange

RSA is an asymmetric cryptographic method based on a public and private key pair [8]. In a cloud sharing scenario, RSA can be used to protect the AES session key. The data owner encrypts the AES key with the recipient's public key,

and only the recipient can decrypt it using the corresponding private key. This approach avoids sending the symmetric key in plaintext.

C. SHA-256 for Integrity Verification

SHA-256 is part of the Secure Hash Standard and produces a fixed-size digest from input data [9]. A key property of cryptographic hash functions is that even a small change in the file produces a very different digest. Therefore, the owner can compute a hash before uploading the file, and the recipient can recompute it after downloading to verify integrity.

D. Digital Signatures for Authenticity

Digital signatures are used to prove that a message or file was created by a specific sender and was not modified after signing. A sender signs data using a private key, and other users verify the signature using the sender's public key. This provides authenticity and non-repudiation, which are important when multiple users collaborate on shared data.

E. Audit Logging for Accountability

Cryptography protects data content, but auditing protects the process around the data. An audit log records actions such as upload, download, update, deletion, sharing, and permission change. In secure cloud environments, logs support accountability and help investigators identify the source of suspicious activity.

VI. Proposed Methodology

The proposed methodology follows a layered protection model. Instead of relying only on the CSP, it places the main security operations on the client side and uses the cloud primarily as a storage and delivery platform.

A. Step 1: Client-Side Pre-Encryption

Before a file is uploaded to the cloud, the data owner generates a random symmetric session key and encrypts the file locally using AES. Only the encrypted file is uploaded to the CSP. This prevents the CSP from reading the original data content, even though it stores the encrypted file.

B. Step 2: Secure Key Sharing

To share the encrypted file with an authorized user, the owner obtains the recipient's public key. The owner then encrypts the AES session key using RSA and sends the encrypted key package to the recipient. The recipient uses the corresponding private key to recover the AES session key and decrypt the shared file.

C. Step 3: Integrity Check

Before uploading the file, the owner calculates a SHA-256 hash of the original file or the encrypted file, depending on the system design. After download, the recipient calculates the hash again and compares it with the trusted original value. If the hashes match, the file is considered unchanged. If they differ, the file may have been corrupted or tampered with.

TABLE II
Proposed Secure Sharing Workflow

Step	Description
1	Encrypt the file locally using AES before cloud upload.
2	Encrypt the AES key with the recipient's public RSA key.
3	Generate and verify a SHA-256 hash for integrity checking.
4	Apply a digital signature for authenticity and non-repudiation.
5	Record sharing and access operations in an audit log.

D. Step 4: Digital Signature Verification

The owner signs the file metadata or hash using a private key. The recipient verifies the signature using the owner's public key. This confirms that the file was shared by the expected owner and that the signed data has not been altered.

E. Step 5: Access Audit

Every access attempt should be recorded in an audit log. The log should include the user identity, timestamp, operation type, file identifier, and access result. This creates an accountability trail and supports forensic analysis if suspicious activity occurs.

VII. Advantages of the Proposed Approach

The proposed methodology provides several practical advantages for secure cloud data sharing.

First, client-side encryption improves privacy because the cloud provider stores only encrypted content. Even if cloud storage is inspected by an internal administrator or compromised by an external attacker, the attacker cannot read the original file without the decryption key.

Second, secure key exchange separates data storage from key management. The CSP may store the encrypted file, but the data owner controls who receives the decryption key. This reduces dependency on provider-side access control alone.

Third, hash-based integrity checking gives users a simple way to detect unauthorized modification or accidental corruption. This is important because cloud users often cannot directly inspect the physical storage system.

Fourth, digital signatures and audit logs strengthen accountability. In multi-user cloud environments, it is not enough to know that data is encrypted; it is also necessary to know who accessed it, when they accessed it, and whether the data was shared by the correct owner.

VIII. Challenges and Limitations

Although cryptographic protection improves security, it also introduces practical challenges.

A. Key Management Complexity

Key management is one of the hardest problems in secure cloud sharing. As the number of users and files increases, the system must manage many encryption keys, public keys, private keys, and access policies. If a private key is lost,

the user may permanently lose access to encrypted files. If a private key is stolen, all data accessible through that key may be compromised [10].

B. Performance Overhead

Encryption, decryption, hashing, and signing require computation. For large files or resource-limited devices, these operations may introduce delay during upload and download. Although AES is efficient, the full secure sharing workflow can still create overhead when repeated frequently or applied to large datasets.

C. Revocation Difficulty

Revocation is difficult because a user may keep a previously received decryption key. To fully revoke access, the owner may need to re-encrypt the file with a new key and distribute the new key to remaining authorized users. This process becomes expensive when many users or files are involved [5].

D. Metadata Leakage

Even when file contents are encrypted, metadata may still reveal sensitive information. File names, sizes, upload times, sharing relationships, and access patterns can expose useful information to attackers. Therefore, secure cloud sharing should also consider metadata minimization and privacy-preserving access mechanisms.

IX. Future Research Directions

Future work in secure cloud data sharing should focus on improving usability and reducing performance costs without weakening security.

One important direction is homomorphic encryption, which allows computation on encrypted data without decrypting it first. Although fully homomorphic encryption is still expensive for many practical workloads, it has the potential to reduce trust in cloud providers for data processing tasks.

Another direction is attribute-based encryption, where access policies are embedded into the encryption process. This can simplify fine-grained access control for organizations with many users and changing roles.

A third direction is privacy-preserving auditing. Public auditing can help verify integrity, but the auditor should not learn the content of the file or sensitive access patterns. Combining auditing with privacy-preserving cryptographic protocols remains an important research area [4].

X. Conclusion

Secure data sharing in cloud computing is a critical problem because cloud users benefit from collaboration and scalability while losing direct physical control over their data. This paper reviewed the main risks of shared cloud storage, including provider trust issues, unauthorized access, and data integrity violations. It also discussed core security requirements such as confidentiality, integrity, authentication, access control, revocation, and accountability.

The proposed methodology combines client-side AES encryption, RSA-based key sharing, SHA-256 integrity verification, digital signatures, and access auditing. This layered approach reduces reliance on the cloud provider and gives the data owner stronger control over shared information. However, practical deployment still requires careful attention to key management, performance overhead, revocation, and metadata leakage.

Overall, secure cloud sharing should not be understood as simply uploading encrypted files. It should be designed as a complete security framework that balances privacy, usability, efficiency, and trust. As cloud systems continue to support larger and more sensitive collaborations, these principles will remain essential for protecting data in modern digital environments.

References

[1] M. Armbrust *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proc. Financial Cryptography and Data Security Workshops*, 2010, pp. 136–149.

[3] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. ASIACRYPT*, 2008, pp. 90–107.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[5] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182–1191, 2013.

[6] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[7] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," FIPS Publication 197, 2001.

[8] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[9] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," FIPS Publication 180-4, 2015.

[10] National Institute of Standards and Technology, "Recommendation for key management: Part 1 - General," NIST Special Publication 800-57 Part 1, Revision 5, 2020.