ISSN : 2350-0743

International Journal of Recent Advances in Multidisciplinary Research

Vol. 13, Issue 05, pp.12455-12459, May, 2026
https://doi.org/10.64485/ijramr.6624.05.2026

REVIEW ARTICLE

A SECURE DATA DYNAMICS AND PUBLIC AUDITING SCHEME FOR CLOUD STORAGE

Pavithra, K. and *Mrs. Gayathri, G.

Dept. of Computer Science and Engineering, Jayalakshmi Institute of Technology, Thoppur, Dharmapuri – 636352
Anna University, Chennai 600 025; Supervisor, M.E., (Ph.D), Associate Professor

ARTICLE INFO

Article History:
Received 15th February, 2026
Received in revised form
24th March, 2026
Accepted 19th April, 2026
Published online 28th May, 2026

Keywords:

Cloud Storage, Public Auditing, AES-256, SHA-512, RSA-15360, Data Dynamics, Third Party Auditor, Integrity Verification.

Corresponding author: Mrs. Gayathri, G.

ABSTRACT

Cloud computing is an evolving technology that provides data storage and highly fast computing services at very low cost. Data stored in the cloud is managed by cloud service providers, raising concerns about authenticity, reliability, and integrity. Unauthorized users may misappropriate or alter data. This paper proposes a secure public auditing scheme employing a Third Party Auditor (TPA) to authenticate the privacy, reliability, and integrity of data stored in the cloud. The proposed scheme uses AES-256 for encryption, SHA-512 for integrity verification, and RSA-15360 for public key encryption, and supports data dynamics operations including insertion, deletion, and modification.

Citation: Pavithra. K. 2026. "A Secure Data Dynamics and Public Auditing Scheme for Cloud Storage***" , International Journal of Recent Advances in Multidisciplinary Research, 13,(05), 12455-12459.***

INTRODUCTION

Cloud computing provides on-demand data storage and sharing services such as Dropbox and Google Drive, enabling groups of users to collaborate by accessing and modifying shared data. Once a user creates shared data in the cloud, every member of the group can access, modify, and share the latest version with others. Although cloud providers promise secure and reliable environments, the integrity of cloud data may still be compromised due to hardware and software failures and human errors. To protect data integrity, various mechanisms have been proposed. A signature is attached to each block of data, and the integrity depends on the correctness of all signatures. A key feature of these mechanisms is to allow a public verifier to efficiently check data integrity without downloading the entire dataset — a process known as public auditing or Provable Data Possession (PDP). The public verifier may be a client wishing to utilize cloud data for computation, search, or data mining, or a Third-Party Auditor (TPA) who provides verification services on behalf of users.

When a user in a shared group modifies a block, she must compute a new signature for that block. Different blocks are signed by different users. For security, when a user leaves a group or misbehaves, that user must be revoked. The blocks previously signed by the revoked user must be re-signed by an existing user. A naive approach requires an existing user to download, verify, re-sign, and re-upload all affected blocks — an expensive operation in terms of communication and computation, especially for large files or frequently changing group memberships.This paper proposes a novel public auditing scheme with efficient user revocation using proxy re-signatures.

Upon revocation, the cloud re-signs affected blocks using a re-signing key. This significantly improves revocation efficiency while preventing the cloud from signing arbitrary blocks on behalf of any user. The proposed scheme employs AES-256 for data encryption, SHA-512 for integrity checking, and RSA-15360 for public key operations, supporting full data dynamics (insertion, deletion, and modification).

LITERATURE REVIEW

Identity-Based Public Auditing with Key-Exposure Resilience: Secured storage is a critical component in cloud computing. Existing cloud auditing schemes based on Public Key Infrastructure face challenges of certificate management and high computation time. Identity-based schemes eliminate certificate usage but limit damage due to key exposure to earlier time periods only. An Identity-based Provable Data Possession scheme with strong key-exposure resilience is proposed to overcome these limitations.

Survey on Data Integrity Auditing in Cloud Computing: Cloud computing enables data outsourcing, allowing data owners to host large datasets and users to access them as required. The prototype of data outsourcing introduces new security challenges since users may not fully trust Cloud Service Providers (CSPs). Many auditing schemes have been proposed to maintain cloud data integrity and address user-CSP trust issues.

Blockchain-Based Public Auditing Without Trusted Auditors: Current auditing schemes rely on a trusted TPA, which cannot fully eliminate the possibility of malicious auditors. Blockchain technology can solve the trust problem among multiple parties. A public auditing scheme using blockchain to resist malicious auditors is proposed, demonstrated to be both feasible and efficient through experimental analysis.

Survey of Public Auditing for Secure Cloud Storage: Cloud storage services offer large storage capacity but raise data security concerns since data is not stored on users' own devices. Public auditability — delegating integrity verification to a TPA — achieves efficiency and security. Basic requirements, evaluation metrics, and representative approaches for analysis of security and efficiency are surveyed, along with future development directions.

Secure Data Sharing with Key Aggregate Cryptosystem: Cloud computing allows consumers and businesses to use applications and access personal files without local installation. Key aggregate cryptosystem supports flexible delegation where any subset of ciphertexts is decryptable using a constant-size decryption key. An overview of cryptographic techniques for efficient secure data sharing in cloud storage is presented.

Authorized Auditing with Auditability-Aware Data Scheduling: Storing data in a cloud server introduces security threats since clients have no direct control after upload. TPA handles auditing as an external trusted entity. Auditability-aware data scheduling handles resource utilization properly, balancing upload, edit, and integrity-checking requests from users and TPAs.

Dynamic and Public Auditing with Fair Arbitration: This work addresses data dynamics support, public verifiability, and dispute arbitration simultaneously. An index switcher preserves mapping between block and tag indices, eliminating passive effects on tag computation. Fairness guarantees ensure neither data owner nor cloud can misbehave in the auditing process.

Cloud Data Auditing Techniques: Privacy and Security Focus: MACs are simple to use in cloud auditing but cause problems in practice. Homomorphic authentication does not require sharing a secret key. BLS (Boneh-Lynn-Shacham) uses bilinear pairing for verification and supports public auditing and data dynamics. Data stored on the cloud comes from devices with different backhaul networks (2G, 3G, LTE, 4G), requiring synchronized auditing architectures.

Privacy-Preserving Public Auditing for Cloud Storage: A privacy-preserving public auditing system enables users to remotely store data and enjoy on-demand high-quality services without the burden of local preservation. Users should be able to use cloud storage as if it were local, without worrying about its dependability. Enabling public auditability via a TPA is of vital importance for users to maintain worry-free outsourced data integrity.

Secured Public Auditing with Dynamic Structure: Cloud computing raises concerns about data security and privacy. Data sharing with sensitive information hiding and remote data integrity auditing is proposed through the concept of identity-based shared data integrity auditing. Data is outsourced to the cloud only after authorization by a proxy, and keys are generated randomly by a Key Generation Centre.

SYSTEM ANALYSIS

Existing System: In the existing system, files uploaded to the cloud are not signed by users on each upload, making integrity verification of shared data impossible. Outsourcing every user's private key to the cloud introduces significant security issues. Traditional cryptographic technologies for data integrity and availability cannot operate on outsourced data.

Downloading data for validation is not a practical solution due to expensive communication overhead, especially for large files. The cloud infrastructure, although powerful and reliable, is susceptible to both internal (loss/destruction of data) and external (unauthorized access) threats.

Limitations of Existing System:

Files not signed by user at each upload. Integrity of shared data is not achievable. Cloud is not in the same trusted domain as users. Outsourcing private keys introduces significant security risks

Proposed System: The proposed system assumes the cloud may lie to verifiers about shared data incorrectness to protect its reputation and avoid financial loss. No collusion between cloud and any user is assumed. Under this semi-trusted model, data incorrectness can be introduced by hardware/software failures or human errors. The Cloud Server allows the TPA to audit data stored in the cloud as requested by users. The TPA is also capable of auditing multiple files simultaneously.

Key Contributions: SHA-512, developed by NIST as part of the SHA-2 family based on the Merkle-Damgård scheme, generates a 512-bit hash value — the longest available — making it highly resistant to attacks.

The proposed system leverages SHA-512 for integrity checks due to its robustness and speed. The generation of encryption parameters using Fully Homomorphic Encryption (FHE) is faster than Paillier encryption. FHE time complexity is $O(\gamma)$, while Paillier encryption complexity is $2 \cdot O(\log N^3)$. Signing time comparisons with DPRDP and DMR-PDP schemes across file sizes of 1 MB, 5 MB, 10 MB, and 20 MB demonstrate that the proposed approach consistently outperforms DMR-PDP in signing efficiency due to elimination of extra multiplication and addition operations.

Advantages of Proposed System

Ability to block compromised user accounts. Security question-based additional authentication. Login with secret key at every session for enhanced security

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It can be seen as the application of systems theory to product development.

Architecture Diagram: The system architecture defines the structure, behaviour, and views of the system. As shown in Fig. 1, data owners encrypt text files using the Advanced Encryption Standard (AES) algorithm to produce an encrypted file and encrypted indexes. These are sent to the Application Server, which communicates with the Semi-Trusted Cloud Server for storage. Data users send search requests through the Application Server, authenticate themselves, and receive ranked results. Data encryption keys are shared between data owners and data users.

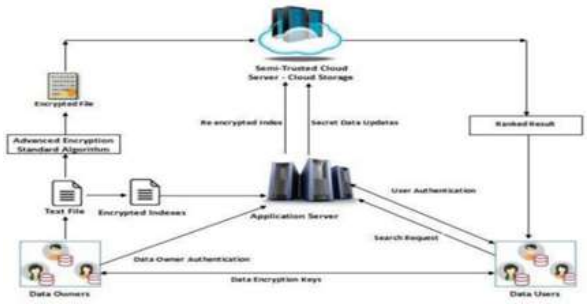

Fig. 1. System Architecture Diagram

Data Flow Diagram: Fig. 2 illustrates the data flow across the four user roles — Patient, Physician, MD, and Cloud. After registration and login, each role accesses its respective functions: the Patient uploads records and grants permissions; the Physician views patient and record information; the MD views records and attacked files; and the Cloud manages key assignment, permissions, and leaked report generation.

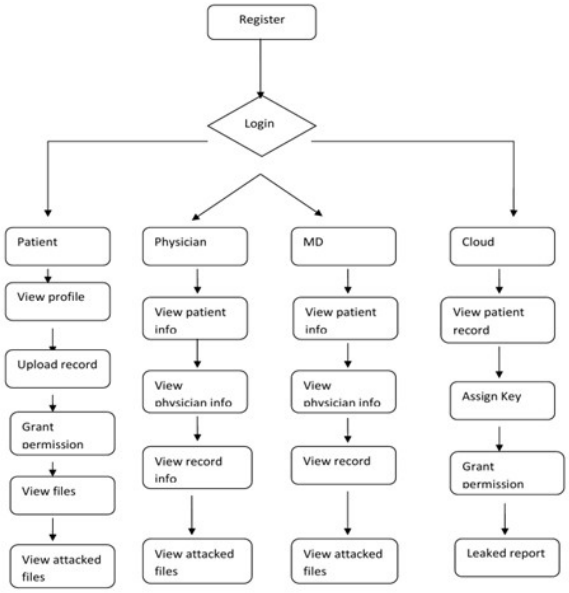

Fig. 2. Data Flow Diagram

Use Case Diagram: Fig. 3 presents the use case diagram showing interactions between actors (Patient, Physician, MD, Cloud) and the system use cases including Register, Login, View Profile, Upload Record, Grant Permission, View Record, View Attacked File, Assign Key, Leaked Report, and Sign Out.

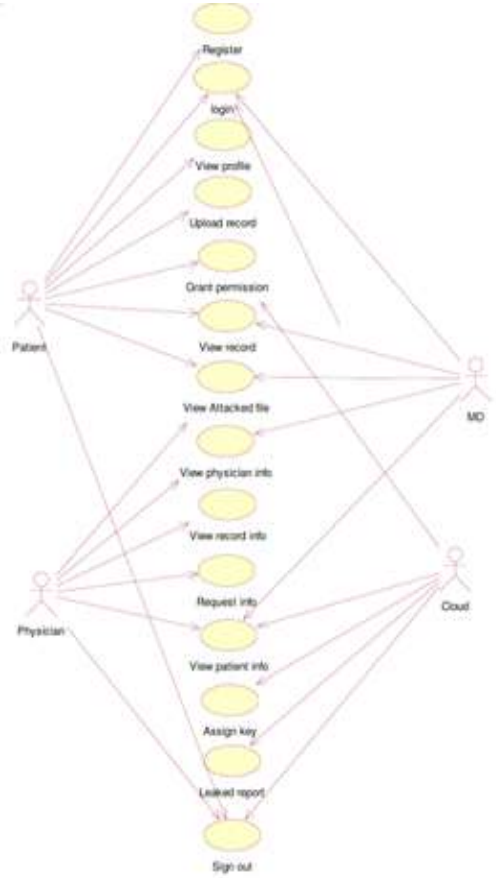

Fig. 3. Use Case Diagram

Class Diagram: Fig. 4 shows the class diagram. The Patient class includes Register and Login operations and methods for Upload Record, Grant Permission, View Record, and View Attacked File. The Physician class inherits from Patient and extends with View Request Info. The MD class handles administrative viewing. The Cloud class manages key assignment, patient records, permissions, and leaked reports.

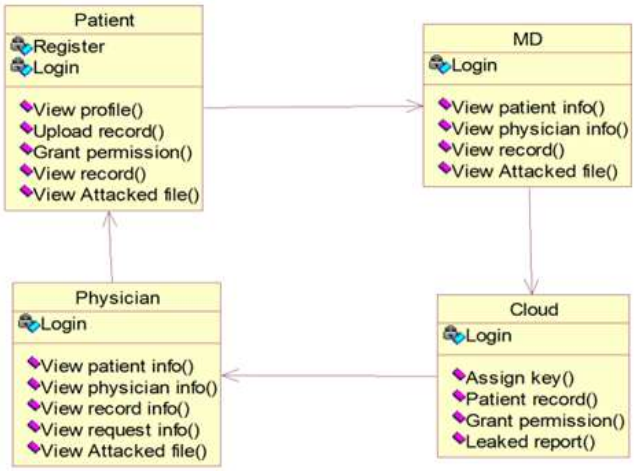

Fig. 4. Class Diagram

Sequence Diagram: Fig. 5 illustrates the sequence of interactions among Patient, Physician, MD, Cloud, and Database. The flow starts with registration, followed by login, profile viewing, record upload, permission granting, file viewing, and detection of attacked files. Each actor logs out after completing their session.

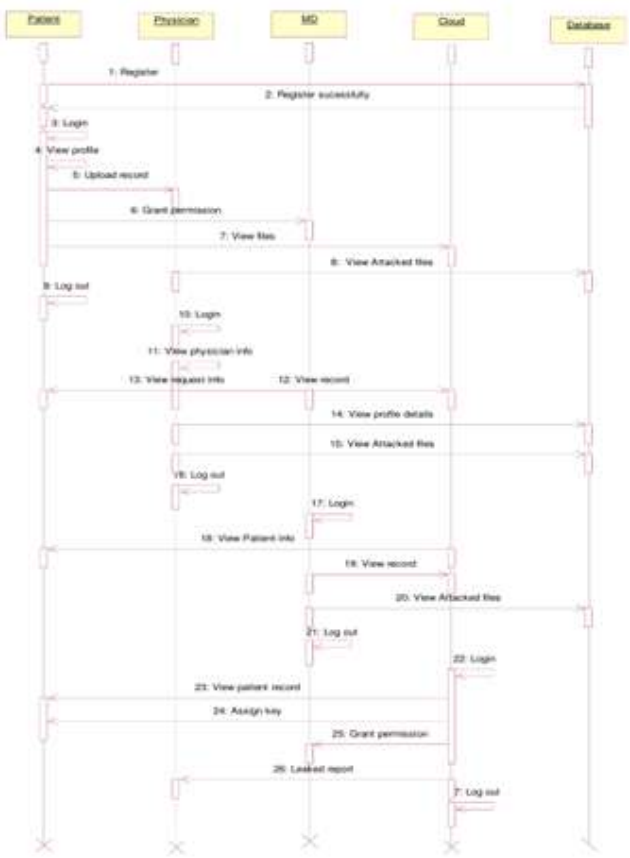

Fig. 5. Sequence Diagram

Activity Diagram: Fig. 6 depicts the activity diagram beginning with user registration. The system branches to four parallel workflows: Patient (upload, grant permission, view attacked files), Physician (view patient/physician/record info), MD (view patient info, records, attacked files), and Cloud (assign key, grant permission, generate leaked report). All workflows converge at Logout.

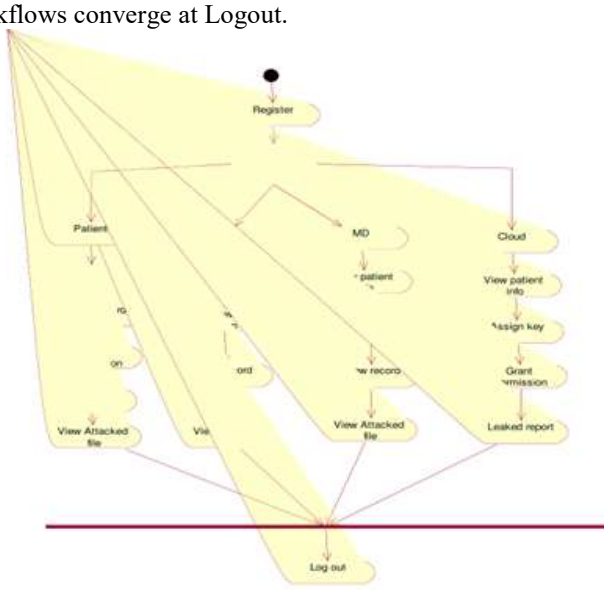

Fig. 6. Activity Diagram

Collaboration Diagram: Fig. 7 presents the collaboration diagram showing numbered message flows among Patient, Physician, MD, Cloud, and Database entities. It captures the complete interaction sequence from registration (1–2) through login, data operations, key assignment, permission granting, leaked report generation, and logout (27).

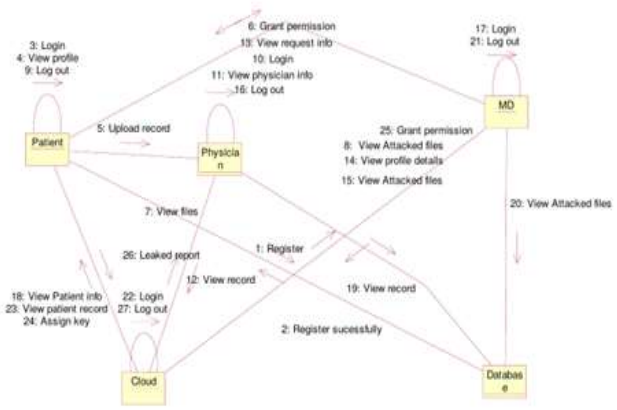

Fig. 7. Collaboration Diagram

SYSTEM IMPLEMENTATION

The system is implemented with the following four core modules:

Data User Authentication: To prevent attackers from impersonating legitimate data users and launching statistical attacks based on search results, users must be authenticated before the administration server encrypts trapdoors. Authentication follows three steps: (1) the data requester and authenticator share a secret key k_0; (2) the requester encrypts personally identifiable information d_0 using k_0 and sends the ciphertext $E(d_0)k_0$ to the authenticator; (3) the authenticator decrypts the received data using k_0 and verifies the identity.

Illegal Search Detection: The authentication process is protected by a dynamic secret key and historical information. If an attacker successfully eavesdrops on the secret key but not the historical data, they cannot correctly construct authentication data, and the illegal action is immediately detected by the administration server.

If an attacker obtains all data, they can correctly construct authentication data. The dynamic key mechanism ensures that each update invalidates previous secrets, significantly limiting the window of vulnerability.

Data User Revocation: Unlike previous works, data user revocation in the proposed scheme does not require re-encrypting or updating large amounts of data on the cloud server. Instead, the administration server only updates the secret data stored on the cloud. Secret keys are randomly regenerated for every update operation, expiring all previous trapdoors. Without the administration server's assistance, a revoked user cannot generate correct trapdoors, preventing further search access.

Matching Different-Key Encrypted Keywords: In practical cloud applications, numerous data owners use their own secret keys to encrypt sensitive data for privacy.

When keywords from different data owners are encrypted with different secret keys, locating matching encrypted keywords across multiple owners becomes a challenge. This module enables secure, efficient, and convenient searches over encrypted cloud data owned by multiple data owners with heterogeneous keys.

REQUIREMENTS

Hardware Requirements

Table I. Hardware Requirements

Component	Specification	Purpose
Processor (CPU)	Intel Core i5/i7 or AMD equiv. (≥2.4 GHz)	Cryptographic operations
Memory (RAM)	8 GB or more	Large data block handling
Storage	500 GB HDD / 256 GB SSD	Data, signatures, audit logs
Network	Stable broadband / LAN	Cloud uploads and auditing
Cloud Server	AWS / Azure / Google Cloud / OpenStack	Storage and re-signing proxy

Software Requirements

Table II. Software requirements

Software	Description / Use
Operating System	Windows 10/11 or Linux (Ubuntu preferred)
Programming Language	Java / Python / C++
Cryptography Libraries	BouncyCastle (Java), PyCryptodome / OpenSSL (Python)
Cloud Simulation	Dropbox / Google Drive API; OwnCloud / NextCloud / OpenStack
Database	MySQL / SQLite / MongoDB
Web Server	Apache Tomcat (Java) or Flask/Django (Python)
IDE	Eclipse / PyCharm / Visual Studio Code
Re-signature Impl.	JPBC (Java) or Charm-Crypto (Python) — bilinear pairings

CONCLUSION

This paper presented a secure data dynamics and public auditing scheme for cloud storage. The proposed system employs AES-256 for strong data encryption, SHA-512 for robust integrity verification, and RSA-15360 for secure public key operations. The incorporation of a Third Party Auditor enables public verification of cloud-stored data without requiring users to download the entire dataset. Proxy re-signature-based user revocation significantly reduces the communication and computation burden on existing users. The system supports dynamic data operations — insertion, deletion, and modification — while maintaining the public auditability property. The implementation of illegal search detection and different-key encrypted keyword matching further strengthens the security posture of the proposed scheme. Future work will focus on reducing auditing overhead through batch verification and extending the scheme to support multi-cloud environments.

REFERENCES

Global cloud computing market report 2019.

Agarkhed J. and R. Ashalatha, "An efficient auditing scheme for data storage security in cloud," in Proc. ICCPCT, 2017.

Saroj, S. K. G. Noida, S. K. Chauhan, and A. K. Sharma, "Threshold cryptography based data security in cloud computing," 2015.

Mell P. and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., 2009.

Morea S. and S. Chaudhari, "Third party public auditing scheme for cloud storage," Int. J. Procedia Computer Science, vol. 79, pp. 69–76, 2016.

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

Adokshaja B. L. and S. J. Saritha, "Third party public auditing on cloud storage using the cryptographic algorithm," in Proc. ICECDS, 2017.

Wang, C. Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, 2010.

Wang, B. B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," IEEE Trans. Services Computing, vol. 8, no. 1, pp. 92–106, 2015.

Yu, Y. M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," IEEE Trans. Information Forensics and Security, vol. 12, no. 4, pp. 767–778, 2017.

****