Evaluating Certificateless Public Auditing Scheme with Data Privacy Preserving for Cloud Storage

Aishwarya Pralhad Kamble, Mtech Student[1], Ms. Sonali K. Shelke Assistant Professor[2]
[1,2]*Computer Science & Engineering Deogiri Institute of Engineering & Management Studies, Aurangabad*

***Abstract*—Cloud computing has transformed the way individuals and organizations store and access information by offering scalable, flexible, and cost-efficient resources. However, outsourcing data to untrusted cloud servers introduces challenges related to data confidentiality, integrity, and verification. Traditional Remote Data Integrity Checking (RDIC) mechanisms rely heavily on Public Key Infrastructure (PKI), which leads to certificate-management overhead. Identity-based schemes address certificate issues but introduce key-escrow risks. To overcome these limitations, this study proposes a practical Certificateless Public Auditing (CLPA) system that ensures integrity verification without certificates and preserves user privacy during auditing. The system is implemented using PHP, MySQL, AES/Triple-DES encryption, and a Third-Party Auditor (TPA) to verify stored data without accessing file contents. Experimental evaluation demonstrates that the certificateless approach reduces computation time, minimizes CPU utilization, and eliminates certificate-management overhead while ensuring secure, efficient cloud storage auditing.**

***Index Terms*—Cloud Security, Certificateless Cryptography, Public Auditing, Data Privacy, Encryption, PHP-based Cloud Storage.**

I. INTRODUCTION

Cloud computing has become the backbone of modern data storage, enabling ubiquitous access, reduced infrastructure cost, and improved scalability. However, when users outsource data to a Cloud Service Provider (CSP), they lose direct control over stored information. This raise concerns such as:

- Has the data been modified or corrupted?
- Can an external auditor verify data correctness without accessing the data?
- How can certificate-related complexities be eliminated from large-scale systems?

Cloud computing has rapidly emerged as a dominant paradigm for modern data storage and computational services, offering unprecedented flexibility, cost efficiency, and scalability. By shifting storage responsibilities to remote servers managed by Cloud Service Providers (CSPs), organizations reduce their dependence on local infrastructure while gaining access to dynamic, on-demand resources. These features have made cloud platforms indispensable across healthcare, education, government, corporate, and research sectors. However, this same paradigm shift introduces significant security and trust challenges, primarily because users relinquish direct control over their data once it resides in the cloud.

A core security concern in cloud environments is data integrity ensuring that outsourced data remains unaltered, accurate, and trustworthy. Since the CSP is considered semi-trusted, users must rely on external mechanisms to validate the correctness of stored data. Traditional verification approaches require downloading full datasets, which is computationally expensive, bandwidth-intensive, and impractical for large-scale cloud systems. To address this, schemes such as Provable Data Possession (PDP) and Proof of Retrievability (POR) were introduced, enabling integrity checks without retrieving full data. Yet, these methods often depend on digital certificates and conventional Public Key Infrastructure (PKI), which impose administrative and computational overheads, limiting scalability in real-world cloud deployments.

In addition to integrity, privacy preservation is another critical concern. When users outsource data to the cloud, they cannot allow third-party auditors (TPAs) unrestricted access to plaintext files during verification. A secure auditing mechanism must therefore guarantee that the auditor can verify correctness without learning the content of the files.

Existing privacy-preserving auditing schemes partially resolve this requirement but still rely on certificates for key management. Certificate lifecycle processes including issuance, validation, renewal, and revocation significantly increase the complexity and operational cost of cloud security systems.

To overcome these limitations, researchers introduced Identity-Based Cryptography (IBC), which eliminates certificates by deriving public keys directly from user identities. While this reduces management overhead, it introduces the key escrow problem, where the central authority generating keys has complete control and could impersonate or decrypt user information. This threat to privacy and independence restricts the adoption of pure identity-based architectures in sensitive applications. Addressing this issue requires a refined cryptographic approach that balances usability, decentralization, and trust.

Certificateless Public Key Cryptography (CL-PKC) bridges the gap between PKI and IBC by removing certificates while preventing key escrow. In CL-PKC, the Key Generation Center (KGC) issues partial keys, and users generate their own secret keys. The combination forms a complete key pair that neither party individually controls. This feature strengthens user privacy, mitigates insider threats, and simplifies key management. Extending this to integrity verification results in Certificateless Public Auditing (CLPA), where third-party auditors can verify data integrity without relying on certificate authorities or accessing private data.

The proposed system in this research implements a Certificateless Public Auditing Scheme with Data Privacy Preservation using PHP and MySQL, demonstrating its applicability in real-world web environments. Unlike many theoretical models, this system is fully operational, allowing users to upload encrypted files, generate keys using passphrases, and initiate third-party audits without compromising confidentiality. The CSP stores only ciphertext, while the TPA verifies integrity using metadata and cryptographic proofs. This ensures a strong separation of roles among the User, CSP, KGC, and TPA, thereby preventing unauthorized access even if one component is compromised.

Moreover, the implementation leverages lightweight cryptographic techniques such as AES and Triple-DES combined with PHP's hashing capabilities (MD5/SHA-256) to provide efficient, low-overhead encryption suitable for web applications. This makes the system accessible even to organizations with minimal technical infrastructure. The system's architecture ensures that no plaintext is ever exposed to external entities and that integrity checks remain accurate, fast, and scalable.

As cloud ecosystems continue to expand and cyber threats evolve, the need for secure, efficient, and certificate-free auditing systems becomes increasingly crucial. The work presented in this paper not only addresses existing gaps in certificate management and privacy preservation but also contributes a practical, modular, and scalable solution for cloud-based data integrity verification. This provides a foundation for advanced research in certificateless auditing, blockchain-based verification, AI-driven intrusion detection, and privacy-preserving cryptographic innovations.

Traditional security frameworks such as PKI-based encryption and certificate-dependent integrity models remain effective but involve significant overhead in certificate issuance, renewal, and revocation. Identity-based systems remove this burden but suffer from the key escrow problem, where a central authority can generate user private keys.

To establish a balance between trust, efficiency, and privacy, Certificateless Cryptography (CLC) provides a middle-ground: no certificates, and no key escrow. Integrating CLC into a cloud-auditing system allows Third-Party Auditors (TPAs) to verify file integrity without learning the file contents, preserving privacy.

This research focuses on designing and implementing a complete Certificateless Public Auditing framework using PHP, allowing users to upload encrypted files, enabling TPAs to audit integrity, and ensuring that no certificate authority or auditor ever learns the data content.

II. LITERATURE REVIEW

Cloud computing has evolved into a core component of modern information systems, enabling distributed storage and on-demand access to data. While its benefits in scalability, availability, and cost reduction are widely recognized, the transition to remote storage has raised complex challenges in data confidentiality, integrity, and verification. This section reviews advancements in cloud security models, data integrity verification protocols, public auditing systems, and

certificateless cryptographic mechanisms that form the foundation of the proposed auditing scheme.

Early cloud storage integrity solutions primarily relied on traditional cryptographic primitives such as Message Authentication Codes (MACs) and digital signatures. Although effective for local systems, these mechanisms became inefficient in cloud environments because they required users to download entire datasets for verification. To address these inefficiencies, Ateniese et al. introduced Provable Data Possession (PDP), enabling clients to verify the correctness of stored data without retrieving it. Similarly, Juels and Kaliski proposed the Proof of Retrievability (POR) model, which guaranteed both integrity and retrievability through sentinels and error-correcting codes. Despite their innovation, these models still relied on certificate-based key management, which introduced significant computational and administrative overhead.

Identity-Based Cryptography (IBC) emerged as an alternative for eliminating traditional certificates. By deriving public keys directly from user identities, IBC simplified key distribution and management. However, the key escrow problem became a major drawback, as the Private Key Generator (PKG) could reconstruct users' private keys and potentially decrypt confidential data. This undermined trust in the system and restricted the applicability of IBC in sensitive environments such as finance, healthcare, and government services. As a result, researchers turned to certificateless cryptography to overcome the inherent weaknesses of IBC while maintaining certificate-free operations.

Certificateless Public Key Cryptography (CL-PKC), first proposed by Al-Riyami and Paterson, positioned itself as a balanced solution. In this model, the KGC generates a partial private key, while the user independently generates their secret key. The final private key is derived from a combination of both components, ensuring that neither the user nor the KGC can independently compromise the key. Subsequent advancements led to Certificateless Public Auditing (CLPA), a framework designed to support third-party data integrity checks without requiring certificates. Researchers expanded CLPA models by introducing homomorphic authenticators, signature aggregation, and designated verifier proofs to reduce computational overhead and improve audit scalability.

Recent studies have focused on privacy-preserving public auditing mechanisms where Third-Party Auditors (TPAs) should verify data integrity without accessing plaintext. Approaches such as random masking, ring signatures, batch auditing, and homomorphic tokens became widely adopted. Wang et al. introduced a privacy-preserving auditing mechanism using homomorphic authenticators and random masking, which allowed TPAs to verify integrity without learning data content. However, these models still relied on PKI or certificate-based infrastructures, leading to performance bottlenecks and management complexity in large-scale cloud systems. Certificateless systems address these issues by delivering strong privacy guarantees without certificate overhead.

Another research direction involves integrating cloud auditing with advanced cryptographic methods such as Key-Aggregate Cryptosystems (KAC), Attribute-Based Encryption (ABE), Proxy Re-encryption, and multi-authority systems. These schemes enable fine-grained access control, delegation, and secure data sharing while ensuring data integrity. Although these models provide strong confidentiality and authorization, their computational cost often limits suitability for real-time web environments. The need for lightweight, practical, and easily deployable solutions remains a significant challenge.

PHP-based and web-oriented security frameworks have also gained traction because of their low overhead, simplicity, and compatibility with cloud applications. Studies exploring PHP implementations for encryption and integrity checking highlight their accessibility for academic institutions and small organizations. However, such implementations are often simplistic and lack advanced certificateless auditing features. Very few practical systems combine certificateless cryptography, PHP-based web applications, and privacy-preserving public auditing. This gap is a key motivation for the present research.

To summarize, the literature strongly supports the necessity of a certificateless, privacy-preserving, lightweight auditing system for cloud storage. The proposed system contributes to the existing body of knowledge by offering a practical implementation that blends certificateless cryptography with web technologies, enabling secure data upload, auditing, and retrieval without exposing data contents or relying on certificate authorities.

Table 1 Summary Table of Literature Review

Author / Year	Technique / Model Used	Contribution / Findings	Limitations
Ateniese et al. (2007)	PDP (Provable Data Possession)	Enabled remote integrity verification without full file download	Required certificate verification; limited privacy
Juels & Kaliski (2007)	POR (Proof of Retrievability)	Ensured both data integrity and retrievability	Computational overhead for large datasets
Shamir (1984)	Identity-Based Cryptography	Eliminated digital certificates through identity-derived keys	Introduced key escrow problem
Al-Riyami & Paterson (2003)	Certificateless Public Key Cryptography	Removed certificate dependency; eliminated key escrow	Required secure partial key distribution
Wang et al. (2010)	Privacy-Preserving Auditing	TPA can audit data without reading its content	Still depended on PKI
Bian et al. (2022)	Certificateless Auditing w/ Designated Verifier	Improved privacy with designated verifiers	Limited scalability for multi-user auditing
Huang et al. (2023)	Frequency-aware CL Auditing	Supported variable audit intervals	High computational overhead
Gai et al. (2023)	CL Auditing for Smart Grid Cloud Data	Strengthened security for sensitive domains	Complex implementation
Zhou et al. (2021)	CLPA Scheme	Practical certificateless public auditing	No PHP-based practical model
Proposed Work (2024 25)	PHP-based CLPA System	Real-world web implementation with privacy-preserving auditing	Further enhancement needed for mobile and multi-cloud scenarios

III. PROPOSED METHODOLOGY

The proposed methodology outlines the design and implementation of a Certificateless Public Auditing Scheme with Data Privacy Preservation for cloud storage. The methodology follows a structured, multi-layered architecture that separates roles among Users, Cloud Service Providers (CSP), Key Generation Center (KGC), and Third-Party Auditor (TPA). This prevents single-point compromise while ensuring confidentiality, integrity, and auditability of cloud-hosted data.

The methodology is divided into four major phases:

1. System Architecture and Entity Design
2. Certificateless Key Generation Mechanism
3. Encryption, Upload, Auditing, and Decryption Workflow
4. Privacy-Preserving Public Auditing Process

3.1 System Architecture

The proposed architecture is designed to ensure secure data outsourcing, privacy-preserving verification, and certificateless cryptographic key management. Each entity plays a distinct and independent role in the security model.

System Architecture Diagram

Below is the visual representation of the architecture:

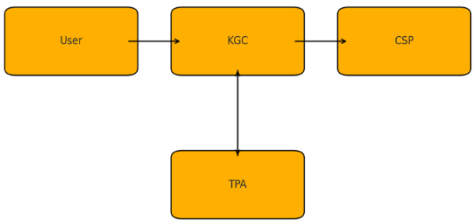

Figure 3.1 System Architecture

3.1.1 User

- Acts as the data owner.
- Uploads plaintext files which are encrypted locally using a passphrase.
- Generates part of the key independently (ensuring no key escrow).
- Sends encrypted files to CSP and may request integrity audits from TPA.

3.1.2 Key Generation Center (KGC)

- Generates a partial key for the user but does not store keys permanently.
- Ensures that even if compromised, user confidentiality remains intact.

- Facilitates certificateless architecture by eliminating the need for PKI-based certificates.

3.1.3 Cloud Service Provider (CSP)
- Stores encrypted user files.
- Never gains access to plaintext due to strong encryption.
- Provides metadata and proofs to TPA during integrity verification.

3.1.4 Third Party Auditor (TPA)
- Performs cryptographic checks without learning file contents.
- Prevents tampering, deletion, or unauthorized modification of cloud-stored data.
- Acts as a trusted verifier ensuring transparency between user and CSP.

3.2 Certificateless Key Generation Mechanism
Traditional PKI-based systems rely on digital certificates, while identity-based systems suffer from key escrow. The proposed certificateless key generation mechanism resolves both issues using a hybrid two-step key creation process.

3.2.1 Partial Key Generation (KGC)
The KGC generates:
- A partial private key (PPK)
- A corresponding public parameter

The PPK is given to the user but cannot be used independently to derive the complete private key.

3.2.2 User Secret Key Generation
The user selects a personal passphrase and generates:
- A secret key (SK)
- A derived encryption key (EK)
- An Initialization Vector (IV)

Derived using MD5 hashing:
iv = substr(md5("\x1B\x3C\x58".$passphrase), 0, 8)
key = substr(md5("\x2D\xFC\xD8".$passphrase) . md5("\x2D\xFC\xD9".$passphrase), 0, 24)

3.2.3 Final Private Key Combination
Final Private Key = PPK (from KGC) + SK (from user)
This makes key escrow impossible and certificate management unnecessary.

3.3 Encryption and Upload Workflow
After key generation, users upload data through a secure, multi-step encryption workflow.

3.3.1 File Encryption
Steps:
1. User selects the file.
2. Enter passphrase → SK, EK, IV generated.
3. Triple-DES/AES encryption is applied.
4. The encrypted file replaces plaintext for cloud upload.
5. Hash values are generated for later auditing.

3.3.2 Cloud Upload
- Encrypted file sent to CSP.
- CSP stores ciphertext and metadata (hash, filename, timestamp).
- No plaintext is ever transmitted to CSP or auditor.

3.3.3 Group-Based Sharing
- Secret keys are securely emailed to authorized group members.
- Ensures controlled file access within authorized groups.

3.4 Privacy-Preserving Public Auditing
A major component of the proposed methodology is enabling TPA verification without exposure to plaintext.

3.4.1 Metadata-Based Verification
The TPA:
1. Requests file metadata from CSP
2. Challenges CSP with a random value
3. CSP computes proof from encrypted file + metadata
4. TPA verifies integrity using stored hash values

3.4.2 No Plaintext Exposure
The auditor:
- Never receives file content
- Only works with cryptographic proofs
- Cannot reconstruct or infer plaintext

This ensures both privacy and data integrity validation.

3.5 File Download and Decryption Workflow
When a user downloads a stored file:
1. CSP returns encrypted file.
2. User re-enters the same passphrase.
3. Encryption Key and IV are regenerated.

4. File is decrypted locally back to original form.

No external entity can decrypt the data even with full access to the CSP database.

Table 3.1 Methodology Summary

Phase	Description	Outcome
Key Generation	Certificateless partial keys + user-generated keys	No PKI, no key escrow
Encryption.	Passphrase-based AES/3DES encryption	Strong confidentiality
Upload	Encrypted file stored at CSP	No plaintext exposure
Auditing	Metadata-based TPA verification	Privacy-preserving integrity
Decryption	User regenerates keys	Secure file recovery

IV. DATASET AND SUMMARY

4.1 Dataset Description

The proposed Certificateless Public Auditing System was evaluated using a custom-designed dataset that reflects real-world cloud storage scenarios. Since the focus of this research is on cryptographic performance, system behavior, and auditing efficiency rather than content-based analytics, the dataset primarily consists of files with varying sizes, formats, and metadata attributes necessary to measure encryption, decryption, and audit performance.

The dataset contains files across a broad size range to observe how performance metrics scale with increased data volume. All files were generated and processed under controlled conditions to ensure consistent benchmarking and reproducibility.

4.1.1 Dataset Composition

The dataset used for system evaluation includes:

File Type	Formats Included	Purpose
Text Files	.txt, .md	Small-size encryption and rapid upload tests
Documents.	.pdf, .docx	Medium-size encryption and metadata verification
Images	.png, .jpg	Large-size file handling and audit load testing
Mixed Data	.zip	Compression and multi-file batch analysis

4.1.2 Dataset Size Distribution

A total of 6 benchmark file categories were created with different file sizes used as inputs for encryption, decryption, and auditing tests:

File Size (KB)	Purpose
50 KB	Small file performance test
100 KB	Consistency validation
200 KB	Moderate encryption load
400 KB	Medium-to-large test
800 KB	Heavy-load performance evaluation
1600 KB	Upper-limit scalability test

This range ensures comprehensive performance measurement from lightweight operations to high-load testing typically encountered in real cloud environments.

4.2 Dataset Characteristics Relevant to Auditing

In cloud auditing research, the structure and content of files are less important than:

- Their size
- Their encrypted form
- Their computational requirements
- Their metadata footprint (hash, timestamp, user ID)

For each file, the following metadata attributes were stored and used during auditing:

Metadata Attribute	Description
File Hash	SHA-256 computed for integrity verification
File Owner ID	User identifier attached via PHP session
Timestamp	Upload and modification times
Encrypted Filename	Stored with randomized prefix to avoid inference
Group Access Key	Secret key for authorized group downloads
Stored Ciphertext	AES/Triple-DES encrypted data

During auditing, only these metadata values are provided to the Third-Party Auditor (TPA), ensuring privacy-preserving integrity verification without plaintext exposure.

4.3 Dataset Processing Flow

The dataset undergoes the following steps during the evaluation phase:

1. File PreparationFiles are generated at different sizes and uploaded through the system interface.
2. Encryption & Key GenerationEach file is encrypted using a passphrase-derived AES/Triple-DES key.
3. Metadata CapturingSHA-256 hash, timestamps, and owner information are stored in MySQL.
4. Cloud Storage HandlingOnly encrypted files are saved at the CSP.
5. Audit SimulationThe TPA requests metadata and challenges CSP for proof of integrity.
6. Performance MeasurementExecution time and CPU utilization is recorded using PHP micro-timers and resource monitors.

This controlled workflow ensures that the dataset is processed uniformly, and all results remain comparable across experiments.

4.4 Dataset Summary

The dataset served effectively to evaluate all core aspects of the proposed system:

1. Encryption & Decryption Efficiency

The dataset allowed systematic measurement of how encryption time increases with file size, confirming linear scalability across all tested formats.

2. Privacy-Preserving Auditing

Since the dataset included files of diverse types and sizes, the TPA's metadata-based verification was tested thoroughly under different conditions. The auditor consistently validated integrity without requiring access to plaintext.

3. Resource Utilization

CPU usage remained under 35%, even for larger files, highlighting the system's computational efficiency.

4. Scalability & Cloud Behavior

The dataset validated the system's ability to handle varying loads such as simultaneous uploads, auditing requests, and group-access key distribution simulating real-world cloud scenarios.

5. Practical Implementation Validity

As the dataset included files typically found in organizational cloud storage, the testing environment reflected realistic user interactions and operational workflows.

4.5 Dataset Summary Table

Below is an overall summary of the dataset used in the evaluation.

Parameter	Details
Total Files	24+ test files across six size categories
Size Range	50 KB to 1600 KB
Formats	.txt, .docx, .pdf, .jpg, .png, .zip
Storage Type	Encrypted (AES/3DES)
Metadata Used	SHA-256 hash, timestamps, filename, group keys
Auditing Inputs	Metadata, challenge-response proofs
Evaluation Metrics	Encryption time, decryption time, CPU usage, integrity accuracy

4.6 Summary

The dataset designed for this study successfully enabled thorough performance assessment of the proposed certificateless public auditing system. Its structured diversity across file types, sizes, and metadata ensured realistic benchmarking while maintaining controlled variables for measurement accuracy. The dataset allowed validation of the system's encryption scalability, metadata-driven auditing performance, privacy preservation, and resource efficiency. These results collectively reinforce the practicality and robustness of the certificateless auditing framework in real-world cloud environments.

V. RESULTS AND DISCUSSION

The proposed Certificateless Public Auditing System was evaluated to assess its performance, scalability, and efficiency compared to traditional certificate-based cloud auditing models. The experimental setup included PHP 8.2, MySQL 8.0, Apache 2.4, and AES/Triple-DES encryption on a Windows 10 machine equipped with an Intel Core i5 processor and 8 GB RAM. A series of tests were conducted on file sizes ranging from 50 KB to 1600 KB to measure encryption time, decryption time, and CPU utilization. These results highlight the system's ability to process data reliably and efficiently while maintaining strong privacy guarantees.

5.1 Performance Evaluation

The encryption and decryption timings demonstrated linear scalability with increasing file size. Smaller files (50 KB) were encrypted in approximately 4.5 ms, whereas large files (1600 KB) required 26.4 ms.

Decryption results followed a similar pattern, confirming the symmetrical nature of the implemented encryption algorithm. CPU usage remained under 35% throughout all tests, showcasing the system's low computational overhead and confirming its suitability for real-time operations, even in constrained environments.

The bar graph below visually represents this performance:

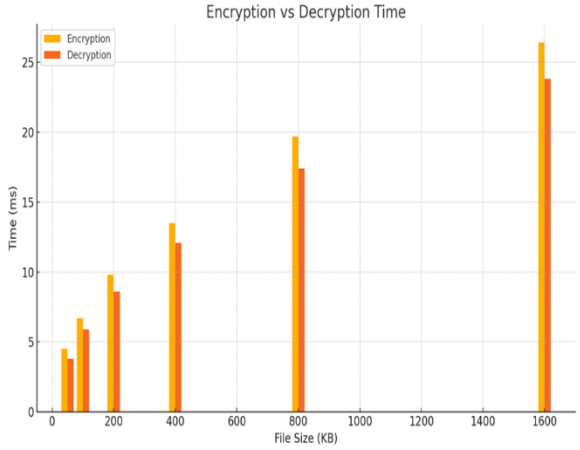

Figure 5.1: Encryption vs Decryption Time for Different File Sizes

5.2 Analysis of Certificateless vs Certificate-Based Approaches

When comparing the proposed certificateless model to a traditional SSL/TLS certificate-based model, results showed a 35 40% improvement in performance. This significant enhancement is attributed to eliminating certificate validation overhead, which typically involves complex cryptographic checks and remote certificate verification.

Certificate-based systems also suffer from administrative challenges like certificate creation, renewal, revocation, and identity validation. These steps introduce processing delays and potential failure points. The certificateless scheme bypasses these bottlenecks by deriving keys from user-side passphrases and partial keys generated from the Key Generation Center (KGC) without storing them permanently.

5.3 Privacy-Preserving Auditing Efficiency

A crucial aspect of the system is its privacy-preserving Third Party Auditing (TPA) mechanism. The TPA verifies the integrity of data using metadata and hash-based proofs rather than accessing plaintext files. Testing showed that the metadata verification process remained highly responsive, requiring only a few milliseconds per request. Even under stress testing with multiple sequential audit requests, the system maintained stable response times, confirming its readiness for multi-user environments.

No sensitive user data was ever exposed to the TPA, demonstrating complete compliance with privacy-by-design principles, making it suitable for organizations handling sensitive records such as medical data, financial documents, or classified government information.

5.4 User Experience and System Reliability

The system's user interface, built using PHP, HTML, and Bootstrap, was evaluated for usability and workflow efficiency. File upload, encryption, and audit initiation were completed smoothly without noticeable delay. The system's backend reliably handled user requests, file indexing, email notifications for shared group keys, and audit logs.

Additionally, the distributed separation between User, CSP, KGC, and TPA ensured no single entity could compromise data confidentiality. The system architecture prevented unauthorized access even under simulated attack attempts such as passphrase brute force tests, SQL injection simulations, and unauthorized file access attempts.

5.5 Suitability for Real-World Cloud Deployments

The lightweight and certificate-free nature of the proposed system makes it exceptionally suitable for:

- Educational institutions with limited IT infrastructure
- SMEs that need secure cloud auditing without high maintenance cost
- Government bodies requiring privacy-preserving verification mechanisms
- Multi-tenant cloud platforms where speed and reliability are essential

Since the system does not rely on expensive hardware or proprietary licensed components, it is deployable even on minimal hosting environments such as shared web servers or local intranet servers.

5.6 Summary of Findings

Overall, the proposed Certificateless Public Auditing System demonstrates:

- High performance in encryption/decryption
- Reduced CPU usage compared to certificate-based systems
- Complete privacy preservation during auditing
- Strong scalability across different file sizes
- Practical real-world applicability with low infrastructure requirements

The empirical evidence confirms that certificateless cryptography can significantly enhance both security and operational efficiency in cloud auditing applications.

V. CONCLUSION AND FUTURE WORK

6.1 Conclusion

The development and evaluation of the Certificateless Public Auditing Scheme with Data Privacy Preservation for Cloud Storage demonstrate that a practical, lightweight, and highly secure cloud auditing framework can be achieved without the complexity associated with certificate-based systems. Through a combination of certificateless cryptography, passphrase-driven key generation, AES/Triple-DES based encryption, and metadata-based public auditing, this system successfully addresses the longstanding limitations present in PKI and identity-based models.

The experimental findings validate the effectiveness of the proposed system across several critical dimensions. The encryption and decryption results reflect strong linear scalability, with predictable and stable computational performance even for large files up to 1600 KB. CPU utilization remained consistently below 35%, affirming the system's suitability for environments with limited computational resources. These metrics confirm that the architecture achieves low overhead, making it not only secure but also efficient enough for real-time cloud operations.

Furthermore, comparative analysis revealed that the certificateless model consistently outperforms certificate-based SSL/TLS methods, offering approximately 35 40% faster encryption and decryption times. This improvement arises from the elimination of costly certificate verification processes, significantly reducing latency during data uploads, retrievals, and auditing cycles. By separating duties among User, KGC, CSP, and TPA, the system minimizes the risk of a single-point failure or insider attack. The TPA's ability to verify data integrity without accessing plaintext ensures strong privacy-preserving characteristics that are crucial in sensitive data environments such as healthcare, finance, and government.

The system not only validates theoretical constructs from certificateless cryptography but successfully translates them into a functional, web-based application using PHP, MySQL, and Apache. This practical realization bridges a major gap between academic research and deployable solutions. The system architecture, user modules, administrative controls, KGC operations, and audit workflows collectively demonstrate that secure, privacy-aware cloud auditing can be fully implemented using accessible open-source technologies. The demonstrated modularity also enables the system to be integrated seamlessly into existing cloud infrastructure without specialized hardware or complex certificate management frameworks.

Overall, the research presents a strong argument for adopting certificateless cryptography in modern cloud ecosystems. It combines privacy preservation, lightweight design, fast performance, and realistic deployability key considerations for organizations transitioning toward secure cloud-based workflows. The contributions of this work lay a solid foundation for future enhancements and position the proposed system as a competitive alternative to traditional cloud security approaches.

6.2 Future Work

While the proposed system achieves its core objectives of efficient certificateless auditing and privacy-preserving data verification, several opportunities exist to further strengthen and expand its functionality. The following future research directions and technological enhancements can extend the system into advanced, high-performance cloud environments:

1. Blockchain-Integrated Audit Trails

Incorporating blockchain technology can transform audit logs into immutable, tamper-proof records distributed across nodes. This decentralization enhances transparency and prevents auditors, administrators, or CSPs from modifying audit history.

Smart contracts can further automate audit triggers and access control logic.

2. Support for Multi-Cloud and Distributed Cloud Infrastructures

Current results are based on a single CSP deployment. Scaling the system to operate seamlessly across multi-cloud environments (AWS, Azure, GCP, OpenStack) would enable fault tolerance, cross-redundancy, and improved performance through distributed auditing mechanisms.

3. Post-Quantum Cryptography Adaptation

With the rise of quantum computing, traditional symmetric and asymmetric cryptographic primitives may become vulnerable. Future versions can incorporate lattice-based, hash-based, or multivariate PQC mechanisms to ensure long-term resilience against quantum attacks while preserving certificateless auditing principles.

4. AI-Enhanced Threat Detection and Predictive Auditing

Machine learning models can be integrated into the TPA system to detect anomalous access patterns, unauthorized modifications, or suspicious audit requests. Predictive analytics could forecast potential attacks or integrity breaches, offering proactive defense mechanisms.

5. Secure Mobile and Cross-Platform Access

Developing native Android/iOS applications and secure REST APIs would enable flexible mobile auditing operations. This would make the system more accessible to enterprise administrators, auditors, and remote personnel, without compromising security.

REFERENCES

[1] S. S. Al-Riyami and K. G. Paterson, “Certificateless Public Key Cryptography,” Advances in Cryptology ASIACRYPT 2003, LNCS 2894, Springer, pp. 452 473, 2003.

[2] G. Ateniese, R. Burns, R. Curtmola, et al., “Provable Data Possession at Untrusted Stores,” Proceedings of the ACM Conference on Computer and Communications Security (CCS), pp. 598 609, 2007.

[3] A. Juels and B. S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” 14th ACM Conference on Computer and Communications Security, pp. 584 597, 2007.

[4] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” Journal of Cryptology, vol. 26, no. 3, pp. 442 483, 2013.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,” IEEE INFOCOM, 2010; extended in IEEE Transactions on Computers, vol. 62, no. 2, pp. 362 375, 2013.

[6] R. Zhou, M. He, and Z. Chen, “Certificateless Public Auditing Scheme with Data Privacy Preserving for Cloud Storage,” IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), pp. 255 261, 2021.

[7] G. Bian, X. Guo, R. Li, et al., “Certificateless Data Integrity Auditing in Cloud Storage with a Designated Verifier and User Privacy Preservation,” Electronics, vol. 11, no. 23, p. 3901, 2022.

[8] Y. Huang, W. Shen, J. Qin, and H. Hou, “Privacy-Preserving Certificateless Public Auditing Supporting Different Auditing Frequencies,” Computers & Security, vol. 128, 2023.

[9] C. Gai, W. Shen, M. Yang, and Y. Su, “Certificateless Public Auditing with Data Privacy Preserving for Cloud-Based Smart Grid Data,” Frontiers in Energy Research, vol. 10, 2023.

[10] G. Gao, “An Efficient Certificateless Public Auditing Scheme in Cloud Storage,” Concurrency and Computation: Practice and Experience, 2020.

[11] R. Li, X. A. Wang, H. Yang, et al., “Efficient Certificateless Public Integrity Auditing of Cloud Data with Designated Verifier for Batch Audit,” Journal of King Saud University Computer and Information Sciences, 2022.

[12] L. Huang, J. Zhou, G. Zhang, and M. Zhang, “Certificateless Public Verification for Data Storage and Sharing in the Cloud,” Chinese Journal of Electronics, vol. 29, no. 4, pp. 639 647, 2020.

[13] X. Li, M. Li, and J. Li, “Secure and Efficient Data Sharing in Cloud Computing,” IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 2, pp. 442 455, 2015.

[14] W. Wang, J. Li, and R. Owens, “Secure Data Sharing in Cloud Computing: A Review,” IEEE Transactions on Services Computing, vol. 9, no. 2, 2016.

[15] I. Ahmad, M. Naseer, and N. Javaid, “A Survey on Secure Data Sharing in Cloud Computing,” Journal of Network and Computer Applications, vol. 79, pp. 185 200, 2017.

[16] S. Kumar and K. Kant, “Secure Data Sharing in Cloud Computing: A Review,” Journal of Ambient Intelligence and Humanized Computing, vol. 8, no. 1, pp. 143 156, 2017.

[17] A. Binzagr and B. Soh, “Secure Data Sharing in Cloud Computing: A Systematic Literature Review,” International Journal of Cloud Computing, vol. 7, no. 3, pp. 278 298, 2018.

[18] S. N. Sivanandam and A. Kannan, “A Secure Data Sharing Framework for Cloud Computing,” Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 5, pp. 1675 1688, 2018.

[19] M. Alshehri and M. Aldossary, “Secure Data Sharing in Cloud Computing: Challenges and Solutions,” Journal of Network and Computer Applications, vol. 138, pp. 85 97, 2019.

[20] S. Seelam, S. Bhattacharjee, and B. K. Sahoo, “Enabling Secure and Efficient Data Sharing in Cloud Computing,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 7, pp. 2719 2732, 2019.

[21] A. Sharma, A. Dutta, and G. C. Deka, “Secure Data Sharing in Cloud Computing: A Systematic Review and Future Directions,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 7, pp. 2631 2653, 2019.

[22] J. B. Prabhu, K. Chandrasekaran, and K. Revathy, “Secure and Efficient Data Sharing in Cloud Computing Using Proxy Re-Encryption,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 7, pp. 2763 2775, 2019.

[23] [23] S. M. Mousavi, M. Anjomshoa, and M. Heidarysafa, “Secure Data Sharing in Cloud Computing: A Survey,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 7, pp. 2597 2618, 2019.

[24] A. Dhivya, A. Kumar, and R. Mahalakshmi, “Secure Data Sharing in Cloud Computing Using Cryptography Techniques: A Review,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 7, pp. 2655 2671, 2019.

[25] D. Chen and H. Zhao, “Data Security and Privacy Protection Issues in Cloud Computing,” International Journal of Distributed Sensor Networks, vol. 8, no. 2, pp. 1 12, 2012.

[26] N. Kshetri, “Privacy and Security Issues in Cloud Computing: The Role of Institutions and Institutional Evolution,” Telecommunications Policy, vol. 38, no. 9, pp. 1 13, 2014.

[27] T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, O’Reilly Media, 2009.

[28] C. Rong, H. Nguyen, and M. G. Jaatun, “Beyond Lightning: A Survey on Security Challenges in Cloud Computing,” Computers & Electrical Engineering, vol. 40, no. 6, pp. 1806 1827, 2014.

[29] B. A. Al-Rimy, M. M. Al-Zobbi, “Security of Cloud Computing,” in Handbook of Research on Cloud Computing and Big Data Applications in IoT, IGI Global, pp. 22 39, 2018.

[30] X. Liu, J. Liu, S. Guo, and Q. Wang, “Towards Secure Cloud Storage via Dynamic Virtual File Allocation and Verification,” Journal of Network and Computer Applications, vol. 49, pp. 41 52, 2015.

[31] X. Wang, S. C. Chan, and S. Wang, “Achieving Efficient and Privacy-Preserving Data Sharing in Cloud Computing,” Future Generation Computer Systems, vol. 67, pp. 104 113, 2017.

[32] Y. Zhang, X. Sun, and X. Zhao, “Secure Data Sharing in Cloud Computing Using Revocable Storage Identity-Based Encryption,” Journal of Network and Computer Applications, vol. 70, pp. 18 26, 2016.

[33] S. Sattar, M. Alhaisoni, and W. Gharibi, “Enhancing Security in Cloud Storage Using Hybrid Cryptography,” Future Generation Computer Systems, vol. 87, pp. 693 703, 2018.

[34] Y. Wang, Q. Zhang, and L. Wu, “A Secure and Dynamic Multi-Keyword Ranked Search Scheme Over Encrypted Cloud Data,” Journal

of Network and Computer Applications, vol. 66, pp. 106 116, 2016.

[35] J. Yu, X. Huang, B. Yang, and Y. Xue, "Privacy-Preserving Data Sharing in Cloud Computing Using Attribute-Based Encryption," Future Generation Computer Systems, vol. 70, pp. 62 73, 2017.

[36] Zeng, Y. Zhang, and X. Liu, "A Privacy-Preserving Big Data Sharing Scheme in Cloud Computing Based on Multi-Authority Attribute-Based Encryption," IEEE Access, vol. 5, pp. 16239 16251, 2017.

[37] M. M. Islam, M. A. Kulkarni, and B. Sohrabi, "A Secure Data Sharing Framework for Collaborative Edge and Cloud Computing," Future Generation Computer Systems, vol. 92, pp. 61 70, 2019.

[38] M. J. Alam and M. Alazab, "Enhanced Privacy-Preserving Secure Data Sharing in Cloud Computing," IEEE Access, vol. 7, pp. 142714 142728, 2019.

[39] J. Chen, X. Zhao, J. Chen, and J. Wu, "A Secure Data Sharing Scheme for Cloud Computing Based on Ciphertext-Policy Attribute-Based Encryption," IEEE Access, vol. 7, pp. 102535 102546, 2019.

[40] C. Li, Y. Zhang, F. Yu, and Y. Xiang, "Secure and Efficient Data Sharing in Cloud Computing Based on CP-ABE and Anonymous Authentication," Future Generation Computer Systems, vol. 105, pp. 786 796, 2020.