Journal of Engineering Technology and Applied Physics

An Improved Authenticator Based Public Auditing Scheme in Cloud Storage

Muhammad Usman*, Maxwell Soita and Mohamed Ahmed Mohamed
Riphah Institute of System Engineering, Riphah International University, G7/4, Islamabad, Pakistan.
**Corresponding author:* usman.s@outlook.com, *ORCiD*: 0009-0002-3333-777X
https://doi.org/10.33093/jetap.2026.8.1.11
Manuscript Received: 20 February 2025, Revised: 23 June 2025, Accepted: 25 November 2025, Published: 15 March 2026

***Abstract*—To minimize costs associated with software maintenance, hardware infrastructure, and secure communication, many organizations and data owners are increasingly opting for cloud storage solutions. However, ensuring the integrity and security of data stored on cloud servers (CS) remains a significant concern. Traditional methods typically use data encryption and decryption to safeguard data integrity, requiring DOs to download, decrypt, audit, and then re-encrypt data before re-uploading it to the CS. This approach is computationally intensive and introduces vulnerabilities, such as data leakage risks, particularly if encryption keys are compromised or data is transferred over insecure channels. While several methods have been proposed to enhance data integrity and availability on cloud platforms, few comprehensively address user data privacy and security, especially in group settings where user revocation is needed. Additionally, current solutions often fail to mitigate risks to sensitive data during the audit process itself. This paper proposes a novel framework designed to protect user identity and data privacy during public auditing of cloud-stored data. By building on and improving existing methods, this framework provides enhanced data confidentiality and integrity while reducing computational and communication overhead. It is also adaptable to various cloud storage models, including Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), thus offering a versatile and practical solution for secure cloud environments.**

***Keywords*—Cloud storage, public auditing, Data integrity, Third-party auditor (TPA), Privacy preservation, Lightweight key generation, Remote data integrity checking (RDIC), Computational overhead.**

I. INTRODUCTION

Cloud storage has revolutionized data management for organizations by reducing costs associated with high-end storage infrastructure, setup, and maintenance. With service models like SaaS, PaaS, and IaaS deployed across hybrid, public, and private models, cloud storage offers scalable and accessible data solutions [1]. However, cloud storage presents risks, notably data breaches, loss, and integrity threats due to malicious attacks and potential Cloud Service Provider abuses. Encryption and decryption were traditionally used to protect data integrity, but they involve high computational costs and present security challenges, such as vulnerabilities in encryption keys [2]. To address these challenges, third-party auditors are increasingly used to verify data integrity without burdening users with computational costs [3]. Mechanisms like Merkle Hash Trees and Remote Data Integrity Checking (RDIC) protocols support public auditing but often rely on complex key management and face key escrow issues. At the same time, solutions like Homomorphic Linear Authenticators and lattice-based privacy preservation have been developed to provide resistance against quantum attacks [4]. Despite advancements, a need remains for efficient, privacy-preserving, and quantum-resistant data auditing mechanisms, especially as quantum computing threatens existing cryptographic approaches [5]. Third-party auditors are semi-trusted as they may be curious about users' personal information, and with powerful computation capacity can deduce users' sensitive information hence during the design of a public auditing mechanism close attention is required to preserve the privacy of user data. Additionally, due to insecure hardware, Trojan viruses etc., user keys may be leaked. Once the user key is compromised an attacker can impersonate to be an authorized owner of data and carry out malicious activities. Moreover, existing public auditing mechanisms are vulnerable to the exponential potential of quantum attacks. The above-mentioned novel mechanism proposed in this paper can guarantee the privacy of user data using cryptography. CS after being compromised can provide a biased audit. Data Owners are not always able to perform audit

Journal of Engineering Technology and Applied Physics (2026) 8, 1, 11:73-83
https://doi.org/10.33093/jetap.2026.8.1.11

Published by MMU PRESS. URL: https://journals.mmupress.com/index.php/jetap/index

operations at regular intervals due to associated computation and communication costs.

This advocates the use of third-party authenticators. However, auditing data with the help of a TPA is still prone to the following hurdles:

- Privacy/ protection of User ID and User Data
- Privacy/ protection of generated keys
- Frequent updates in User Data (Support for dynamic operations)
- Associated Computational Overheads
- Invalid User Revocations

Secure Access of user data who has been revoked. Invalid users intentionally delete or modify data to render it useless. Singh *et. al.* [6] proposed mechanism for secure public auditing using third party trusted data managers, which ensures verification of integrity. It also provides support for dynamic operations. Remote integrity checking and verification has been one of the most effective mechanisms for integrity auditing, but most of these mechanisms are dependent on Public Key Infrastructure (PKI) having substantial communication and storage overheads. Similarly, identity based remote integrity checking and verification has reduced computational overhead costs but face key escrow problem.

The proposed scheme tackles security problems in SaaS, PaaS, and IaaS as presented in Table I. The proposed scheme delivers privacy-protected auditing for SaaS whereas it provides PaaS with lightweight integrity monitoring and IaaS with effective real-time verification thus enhancing cloud security while improving efficiency.

Table II presents an overview of the academic databases used for literature review which includes their areas of coverage and entry and exemption specifications. The research includes IEEE Xplore together with ACM Digital Library and SpringerLink as well as PubMed and Scopus and Web of Science and arXiv and Google Scholar and Elsevier ScienceDirect. The research focuses on peer-reviewed studies with high impact and data science with AI components. It ignores non-peer-review. The methodology establishes a specific approach to literature selection which produces focused reliable research materials.

In this paper, we introduce cloud computing and its security challenges, followed by a literature review covering the current state of cloud security, key threats, and existing security frameworks. The research methodology explains our approach, including how data was collected and analyzed. In the findings section, we discuss the main security challenges and evaluate current auditing frameworks, while the discussion offers insights into these findings and suggests improvements for cloud security practices. The paper concludes with a summary of the main points, a restatement of the thesis, and the importance of the findings, followed by a comprehensive list of references.

Table I. Cloud Storage Models: Definitions, security challenges and our proposed solution.

Cloud Storage Model	Definition	Security & Auditing Challenges	Contribution of the Proposed Scheme
Software as a Service (SaaS)	Provides users with access to cloud-hosted applications without managing underlying infrastructure (e.g., Google Drive, Dropbox, Microsoft OneDrive).	i. Users rely on cloud providers for data security. ii. Risk of data breaches and unauthorized access. iii. Limited user control over auditing mechanisms.	i. Ensures privacy-preserving auditing, preventing Third-Party Auditors (TPA) from accessing user data. ii. Reduces computation overhead for integrity verification. iii. Supports dynamic updates, ensuring data integrity without excessive reprocessing.
Platform as a Service (PaaS)	Provides a cloud environment for developers to build, deploy, and manage applications without handling infrastructure (e.g., Google App Engine, AWS Lambda).	i. Security concerns related to data integrity during application processing. ii. Need for efficient data verification mechanisms for stored application logs and databases.	i. Lightweight auditing ensures secure data integrity checks for PaaS applications. ii. Protects application logs and stored data from unauthorized modifications. iii. Eliminates need for costly key management, improving efficiency.
Infrastructure as a Service (IaaS)	Offers on-demand computing resources like virtual machines, storage, and networking (e.g., AWS EC2, Microsoft Azure, Google Compute Engine).	i. Large-scale storage requires efficient integrity verification. ii. Risk of insider attacks or unauthorized data deletion. iii. High computational cost for continuous auditing.	i. Single-key pair mechanism reduces computational complexity in large-scale data auditing. ii. Ensures tamper-proof auditing, preventing malicious modifications. iii. Supports real-time integrity verification, enhancing trust in IaaS-based storage.

Table II. Descriptions of inclusion and exclusion criteria.

Database	Description	Inclusion Criteria	Exclusion Criteria
IEEE Xplore	A leading database for technical literature in electrical engineering, computer science, and data science.	- Peer-reviewed articles	- Non-peer-reviewed articles
		- Recent publications (last 5 years)	- Outdated papers (more than 10 years)
		- AI and data science-related papers	
ACM Digital Library	A comprehensive resource for computing and information technology fields, including data science and AI.	- Conference proceedings	- Non-indexed papers
		- High-impact data science publications	- Studies not focused on AI, machine learning, or data science
		- Scopus-indexed	
SpringerLink	Offers a wide range of academic publications in data science, AI, and machine learning.	- Journals with impact factor	- Papers outside the scope of AI, data science, or interpretability issues
		- Recent AI and data science papers	
		- High relevance to research question	
PubMed	A database primarily for life sciences and biomedical fields, useful for healthcare-related AI studies.	- Studies focused on AI applications in healthcare	- Non-AI related biomedical studies
		- Peer-reviewed and recent	
Scopus	A large abstract and citation database covering interdisciplinary areas, including data science and AI.	- Indexed journals	- Non-indexed journals or grey literature
		- Studies with a clear focus on neuromyotonic AI, interpretability, and risk assessment	
Web of Science	A high-quality database covering multiple disciplines, widely used for academic research in AI, machine learning, and data science.	- High-impact factor journals	- Articles from non-reputable sources
		- Peer-reviewed articles	- Non-peer-reviewed and low-impact papers
		- Data science and AI relevance	
arXiv	A preprint repository often used for early access to cutting-edge research in AI, machine learning, and data science.	- Preprints with high citations or institutional backing	- Unverified preprints with no peer-review
		- Relevant to neurosymbolic AI and risk assessment	- Studies not directly related to data science or AI
Google Scholar	A broad search engine for academic papers across many disciplines, including AI, machine learning, and data science.	- Peer-reviewed articles	- Non-academic content or grey literature
		- High citation count	- Unverified sources
		- Relevant conference papers or journals	
Elsevier ScienceDirect	A high-quality publisher offering a wide range of journals and articles in AI, machine learning, and data science.	- Articles from high-impact journals	- Non-relevant fields (e.g., humanities, social sciences)
		- Data science, AI, and interpretability-related studies	

II. LITERATURE REVIEW

While many of the proposed and existing mechanisms emphasized the security of data stored on clouds, some gave a fair share to user information privacy. Shacham *et al.* [7] addressed the preservation of privacy by proposing "Homomorphic Linear Authenticator" abbreviated as using masking at random but it failed to preserve the identity privacy of the signers who were introduced to preserve privacy in the first place [8] proposed, employment of a ring signature for data verification which could ensure data integrity without exposing the identity of the signer. Yu *et al.* [9] improved the auditing protocol proposed in [10] to perfection with resistance to key exposure. The proposition's performance was better in key updates outsourcing. An index switcher was proposed by [11] to identify the link between data blocks and indices to evade recalculation of tags which is required in update operations for data blocks, but the index switcher changed sporadically, resulting in substantial extra costs. Yu *et al.* [3] suggested a "Merkle Hash Tree" based mechanism, which supported full dynamic data auditing. More verification and auditing methods for data stored on the cloud which also

supported dynamic operations were proposed in [12 – 16]. Shen *et al.* [16] suggested, auditing based on identity which supported privacy preservation enabling privacy against the auditor by giving the auditor access to zero knowledge. Singh *et al.* [17] proposed a mechanism for secure public auditing using third-party trusted data managers, which ensures verification of integrity using Advanced Encryption Standard (AES)-256 for encryption, Secure Hash Algorithm (SHA)-512 for integrity verification, and Rivest-Shamir-Adleman (RSA)-15360 for generated keys encryption. It also supports dynamic operations (i.e. insertion, deletion, and modification). For encryption and decryption, AES-256 is the most suitable and widely accepted encryption mechanism that can process an input 128 bits input size. There are 2256 possible combinations of 78 digits. This algorithm generates exponentially large numbers and is the strongest encryption algorithm among all existing algorithms. It will take a computer capable of breaking 1 trillion decryptions per second, 257 years to break AES-256. AES-256 is considered more secure than its previous predecessors as mentioned in Table III "SHA-512" was developed by NIST (National Institute of Standards and Technology). It is a member of "SHA-2". It is an updated version based on Merkle-Damaged algorithm. "SHA-512" uses one-way hash function, an expansion of SHA-1 and Message Digest (MD)-4 based improvement.

Table III. Average time key search if search is exhaustive.

Symmetric Cipher	*Number of alternative keys [18]*	*Time required to break cipher if computer do 10^{12} decryption/s [18]*
DES-56	2^{56}	1 hour
1AES-128	2^{128}	5.3×10^{18} year
Triple DES-168	2^{168}	5.3×10^{30} year
AES-192	2^{192}	5.3×10^{37} year
AES-256	2^{256}	5.3×10^{57} year

Table IV. Comparison of security.

Symmetric Cipher	*Equivalent Asymmetric Cipher*	
	RSA	ECC
Skipjack-80	1024	160
Triple DES-112	2048	224
AES-128	3072	256
AES-192	7680	384

Table V. Difference in variation of SHA algorithms.

Algorithm	*Word*	*Message*	*Block*	*Digest*	*Security*
SHA-224	32	$< 2^{64}$	512	224	112
SHA-256	32	$< 2^{64}$	512	256	128
SHA-384	64	$< 2^{128}$	1024	384	192
SHA-512	64	$< 2^{128}$	1024	512	256

Table IV demonstrates that symmetric schemes equate to asymmetric standards yet asymmetric keys have to be substantially larger than symmetric keys to attain equivalent protection. The security levels offered by AES-128 correspond to RSA keys with 3072 bits alongside Elliptic Curve Cryptography (ECC) encryption with 256 bits illustrating the efficiency of ECC regarding shortened key lengths. The dependability of "SHA-512" is based on its ability to generate 512 bits' hash value which is more impervious as it is the longest hash value a hash function can generate. Hence "SHA-512" is most robust, fast and powerful hash function. The Table V shows a comparison between SHA algorithms which includes word size together with message capacity, block size and digest length and security level. The two hash algorithms SHA-224 and SHA-256 process 32-bit words through 512-bit blocks but SHA-384 and SHA-512 operate through 64-bit words with 1024-bit blocks. The security increases with longer digest lengths because SHA-512 provides the maximum strength of 256 bits.

Public auditing of cloud-stored data operates in two main phases. In the first phase, users generate tags associated with their data, which are then uploaded with the original data. The second phase involves a challenge-response system where the auditor checks the integrity of the data, and the results are shared with the user. Traditional public auditing mechanisms based on Public Key Infrastructure often incur additional computational overhead due to key management. However, identity-based public auditing schemes have emerged that significantly lower user computation costs during the tagging phase. Vamshi *et al.* [19] and Zhou *et al.* [20] proposed using a semi-trusted Third-Party Auditor alongside an encrypted keyword search to protect sensitive information. Their method employs a Relation Authentication Label, which authenticates the relationship between files without exposing sensitive data.

Remote integrity checking and verification have proven effective, but many systems still depend on Public Key Infrastructure, leading to high communication and storage costs related to certificate management. Yan *et al.* [21] developed a certificate-less cryptosystem allowing TPAs to conduct integrity audits without expensive key management or escrow issues. Their lightweight design restricts TPAs from analyzing user data while still enabling dispute resolution. Yogita *et al.* [22] introduced a mechanism that binds data to its owner during the proof generation phase, ensuring TPA cannot link the data to its owner, thus preserving privacy. Additionally, Modified Elliptical Curve RSA with a modified MD5 algorithm for attribute-based public auditing enhance security through public key encryption and homomorphic algorithms. As quantum computing advances, existing auditing mechanisms face new security challenges. Fan *et al.* [23] presented a lattice-based technique, LB-PPFS, designed to be resistant to quantum attacks, ensuring data privacy by masking original data and preventing key exposure. Mante *et al.* [24] highlighted the need for efficient auditing methods, introducing multi-searchable attribute-based encryption (VMKS-ABE) to reduce computational costs. This literature indicates a growing focus on developing efficient

public auditing mechanisms that ensure data integrity while minimizing computational overhead, guiding our design for a lightweight key generation model to enhance cloud auditing efficiency.

In the last decade much research [25 - 27], as discussed in the related work section, focused on designing public auditing mechanisms that could ensure the integrity of data and are efficient at the same time. Some of these proposed mechanisms are novel while others are based on previous works. We will design an auditing mechanism based on the proposed lightweight key generation to reduce associated computational overhead. We will ensure the security of stored data remains intact and compare proposed with existing state-of-the-art.

The Table VI provides essential questions for research together with their purposes in developing lightweight key generation methods for public cloud storage auditing systems. This paper looks at how security can be matched with efficiency performance and explores ways to speed up cloud data verification as well as evaluates one key pair against multiple key pairs. The evaluation of the proposed method also carries out a comparison with existing methods based on their computational and communication requirements and storage impacts.

The Table VII shows how different articles address research questions while showing their individual contributions. The research analysis incorporates investigations about lightweight cryptographic key creation for cloud auditing combined with efficiency enhancements and security maintenance. The comparisons evaluate the computational expenses as well as storage requirements of using one key pair versus multiple pairs and measure how well the auditing system performs relative to existing technique.

Table VI. Research questions.

No.	Research Questions	Motivations
1	How can we design a public auditing mechanism for cloud storage that uses lightweight key generation to reduce computational overhead?	The computational cost of generating multiple key pairs can slow down auditing processes. By designing a mechanism with lightweight key generation, we aim to reduce these costs and make cloud auditing faster and more resource efficient.
2	Can the proposed lightweight key generation approach maintain strong data security and integrity while improving efficiency?	Reducing computational overhead is important, ensuring that data security and integrity are not compromised is crucial. This question explores whether lightweight key generation can strike a balance between efficiency and security.
3	How does the proposed auditing method improve the speed and efficiency of cloud data verification compared to existing techniques?	Cloud data verification can be time-consuming and resource heavy. Investigating how the new auditing method speeds up this process without compromising accuracy will help assess its overall effectiveness.
4	What are the benefits of using a single key pair in the auditing process versus multiple key pairs in terms of computation, communication, and storage costs?	Using a single key pair instead of multiple ones simplifies the system. This question seeks to understand how this simplification reduces the overhead involved in computation, communication, and storage during cloud auditing.
5	How does the new auditing mechanism perform against current state-of-the-art methods in reducing overall computational, communication, and storage overheads?	To evaluate the practicality of the new mechanism, it is essential to compare its performance with existing approaches and determine if it truly offers lower overhead while maintaining security.

Table VII. Relevancy of research questions and citations

Research Question	Relevant Article (Title)	Relevance to the Research Question
1. How can we design a public auditing mechanism for cloud storage that uses lightweight key generation to reduce computational overhead?	Article 1: *Lightweight Public/Private Auditing Scheme for Resource-Constrained End Devices in Cloud Storage* [26]	The articles focus on creating more efficient auditing systems through lightweight or dynamic approaches, addressing key generation and overhead reduction, directly answering this question.
	Article 2: *Dynamic Outsourced Auditing Services for Cloud Storage Based on Batch-Leaves-Authenticated Merkle Hash Tree* [8]	The articles focus on creating more efficient auditing systems through lightweight or dynamic approaches, addressing key generation and overhead reduction, directly answering this question.
	Article 3: *Enabling Cloud Storage Auditing with Verifiable Outsourcing of Key Updates* [9]	The articles focus on creating more efficient auditing systems through lightweight or dynamic approaches, addressing key generation and overhead reduction, directly answering this question.

2. Can the proposed lightweight key generation approach maintain strong data security and integrity while improving efficiency?	Article 4: *Identity-Based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage* [3]	These articles deal with maintaining security and integrity in cloud auditing systems, with methods to ensure both privacy and efficiency, which is directly relevant to the proposed key generation method.
	Article 5: *Privacy-Preserving Public Auditing for Shared Data in the Cloud* [11]	These articles deal with maintaining security and integrity in cloud auditing systems, with methods to ensure both privacy and efficiency, which is directly relevant to the proposed key generation method.
	Article 6: *Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud* [28]	These articles deal with maintaining security and integrity in cloud auditing systems, with methods to ensure both privacy and efficiency, which is directly relevant to the proposed key generation method.
3. How does the proposed auditing method improve the speed and efficiency of cloud data verification compared to existing techniques?	Article 7: *Scalable and Efficient Provable Data Possession* [25]	These articles present methods to improve the speed and efficiency of cloud data verification and are used for comparison to assess the proposed auditing method.
	Article 8: *Compact Proofs of Retrievability* [29]	These articles present methods to improve the speed and efficiency of cloud data verification and are used for comparison to assess the proposed auditing method.
	Article 9: *Achieving Efficient Cloud Search Services: Multi Keyword Ranked Search Over Encrypted Cloud Data Supporting Parallel Computing* [30]	These articles present methods to improve the speed and efficiency of cloud data verification and are used for comparison to assess the proposed auditing method.
4. What are the benefits of using a single key pair in the auditing process versus multiple key pairs in terms of computation, communication, and storage costs?	Article 10: *Incentive and Unconditionally Anonymous Identity-Based Public Provable Data Possession* [13]	These articles compare identity-based auditing mechanisms, highlighting different approaches to key pair usage in terms of computational, storage, and communication efficiency.
	Article 11: *Enabling Identity-Based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage* [31]	These articles compare identity-based auditing mechanisms, highlighting different approaches to key pair usage in terms of computational, storage, and communication efficiency.
5. How does the new auditing mechanism perform against current state-of-the-art methods in reducing overall computational, communication, and storage overheads?	Article 12: *Efficient Integrity Auditing for Shared Data in the Cloud with Secure User Revocation* [32]	These articles compare different auditing mechanisms and techniques for reducing computational and storage overheads, making them ideal for evaluating the performance of the proposed mechanism.

III. RESEARCH METHODOLOGY

Cloud storage is now a deeply entrenched practice in our society. There are two main phases in data integrity auditing: The tagging phase and the Proof phase. In the beginning, the user creates tags associated with the data, uploads the data and tags, and the cloud server responds with a challenge. In Proof, the auditor verifies the validity of the response and verifies that the data passed the audit. Cloud storage has become a common practice in our society, but there are still many issues and challenges that need to be addressed. For example, how do users verify the integrity of outsourced data? Especially when they may not have a complete copy of their systems and devices. Some potential solutions to this challenge have been discussed in the literature, including the Provable Data Possession model and Proof of Retrieval model. If the data passes an audit, the user does not need to retrieve it. Furthermore, data integrity auditing is extensively classified. Depending on whether public verifiability is supported, auditing is divided into public and private auditing. Auditing by a third-party Schemes like those shown in [1] and [33] are more efficient. However, unlike public auditing schemes, only the owner can audit the data in private auditing systems, and the judge cannot intervene when a dispute arises. As a result, public auditing techniques are more practical in general. Many public auditing techniques [1, 30, 34 – 36] are built on traditional public key infrastructure-based cryptography. However, one major disadvantage of Public Key Infrastructure-based auditing techniques is the high cost of certificate management, as the schemes rely on certificates provided by a trusted third party to bind the user's identity and public key.

A. Problem Statement

In the last decade much research [25], as discussed in the related work section, focused on designing public auditing mechanisms that could ensure the integrity of data and be efficient at the same time. Some of these proposed mechanisms are novel while others are based on previous works. In Privacy-Aware and Hash-Parity-bits-based public auditing, data is encrypted using hash functions and saved in authenticators in a classical hash tree structure. Through this storage arrangement DO can perform data integrity audits without downloading whole data from CS. To protect data against invalid access and

corruption (update delete to render data useless) by attackers the PAHPPA (Public Health and Preventive Health Policy Act) uses multiple keys for encryption. This arrangement increases the security of data stored on the cloud but on the downside, the additional computational overhead associated with the generation and maintenance of multiple keys renders the proposed auditing mechanism slow and dynamic operations even slower. We will design an auditing mechanism based on the proposed lightweight key generation to reduce associated computational overhead. We will ensure the security of stored data remains intact and compare proposed with existing state-of-the-art.

B. Development Cycle

To achieve these objectives, the research will employ a systematic methodology: Figure 1 depicts the Development Cycle, highlighting the key phases: Algorithm Development, Security Integration, Comparative Analysis, Experimental Validation, and Expected Contribution. This iterative process ensures a robust, practical, and impactful methodology by refining algorithms, evaluating performance, and validating results in real-world scenarios. Design and develop algorithms for the public auditing mechanism, with a special emphasis on optimizing key generation for reduced computational overhead. The new system aims to use less computing power compared to the present method. A lightweight single key pair is the first step towards it, which is the most efficient pair device. It will verify data integrity and availability. It will secure its privacy.

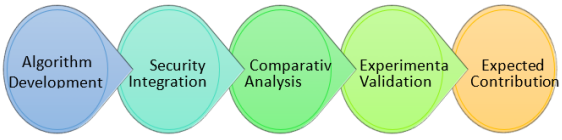

Fig. 1. Development cycle.

The result is a better auditing system with lower initialization and key generation computational cost, which lowers total overhead in computing, communication, and storage. During the initialization and key generation phases, a lightweight single key pair is utilized rather than computationally demanding many key pairs.

C. Improved Auditing System

To improve the effectiveness of data integrity and availability verification in cloud storage, the suggested methodology incorporates an advanced auditing system. In contrast to earlier methods that necessitate downloading and decrypting the entire dataset, our technology lessens the computational load on the data owner. Two important steps in the auditing process are initialization and key generation. The suggested system makes use of creative tactics.

D. Research Objectives

The focus of this research will be the design of an improved public auditing method with lightweight key generation. Our research objectives are as follows: -

- Public auditing mechanism with lightweight key generation
- Ensure that the security of data is not compromised after achieving the first objective.
- Proposition comparison with existing up-to-date mechanisms in terms of computational overhead.

IV. RESULT ANALYSIS

Cloud-stored data can be quickly exchanged and altered by users, but its integrity is at risk from software, hardware, and human errors [2, 37]. To address this, some auditing techniques allow third-party verifiers to check data integrity without requiring users to download entire datasets [38]. This work introduces a privacy-aware public auditing approach that employs homomorphic verifiable group signatures, requiring at least the group members to collaborate in generating a trace key, preventing abuse of authority, and protecting against false accusations among users. Data stored in the cloud is susceptible to loss due to various errors, and cloud owners might downplay incidents to safeguard their reputations [2, 38]. Users often delegate data verification to third-party auditors through public auditing, which risks exposing private information [4]. To mitigate this, previous research has focused on ensuring data and identity privacy using techniques like homomorphic authenticable ring signatures, which maintain user anonymity [34, 35]. Most public auditing methods suffer from single-user tracing issues and lack robust authentication between auditors and the cloud [30]. While Shen et al. developed an authorized public auditing scheme for single clients, it does not extend to group data [36]. Efficient handling of user revocation is crucial, as it often requires reassigning authenticators and can incur significant computational and transmission costs [31, 39]. This study aims to enhance public auditing mechanisms by streamlining processes, optimizing algorithms, and minimizing computational overhead while ensuring user privacy during the auditing process [40]. Ultimately, the proposed method seeks to establish a new standard for secure, efficient public auditing in cloud environments, balancing resource efficiency with robust security measures. Our secure system operates through a cloud environment to provide group data sharing functionality as depicted in Fig. 2.

The Group Manager establishes group enrolment at the Private Key Generator to obtain authorization information needed for Cloud database sharing among authorized members. The system operates user revocation through its mechanisms while the Third-Party Auditor conducts independent verification of data integrity by employing auditing challenges alongside proof mechanisms to guarantee system security and reliability.

Fig. 2. Real-world applicability.

A. Computational Overhead Trend

In general, the computational overhead across most phases tends to increase with an increase in the number of data objects for both implementations.

B. Multiple Key Pair vs. Single Key Pair

Across most phases and across different numbers of Data Objects, the Single Key Pair implementation showcases lower computational overhead compared to the Multiple Key Pair implementation as shown in Fig. 3 and Fig. 4.

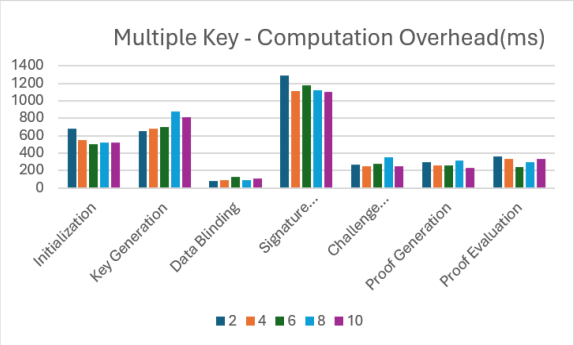

Fig. 3. Multiple key.

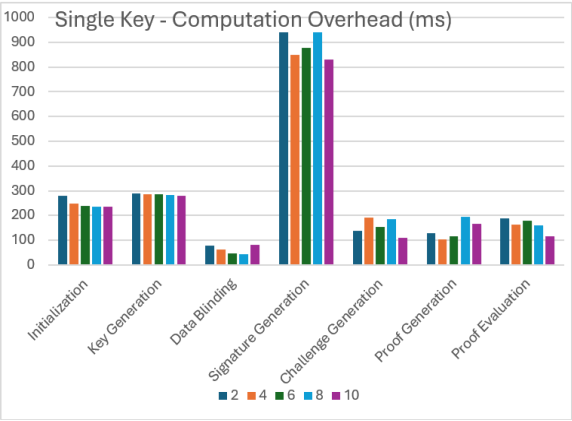

Fig. 4. Single key.

This approach uses one key pair (a public and private key) for all data objects. It consistently has lower computational overhead compared to the Multiple Key Pair approach in most phases, such as Initialization, Key Generation, and Data Binding. The Single Key Pair implementation also requires less storage space compared to the Multiple Key Pair implementation. This approach uses a separate key pair for each data object. It has higher computational overhead in phases like Signature Generation, Proof Generation, and Proof Evaluation, especially when the number of Data Objects increases.

The Multiple Key Pair implementation requires more storage space compared to the Single Key Pair implementation, particularly in phases like Data Binding and Signature Generation. The Single Key Pair implementation is generally more efficient in terms of both computational resources and storage requirements. However, there are some phases where the Multiple Key Pair implementation might be more suitable, depending on specific use cases.

C. Storage Overhead Multiple Key Pair Implementation

Increased storage overhead in the majority of stages when compared to single-pair solution. Significant storage resources are constantly required for phases like Data Binding and Signature Generation. Figure 5 displays a rising trend in the majority of phases and fluctuating storage needs as the DO count rises. As the number of DOs increases, some phases, such as Signature Generation, show a significant increase in storage.

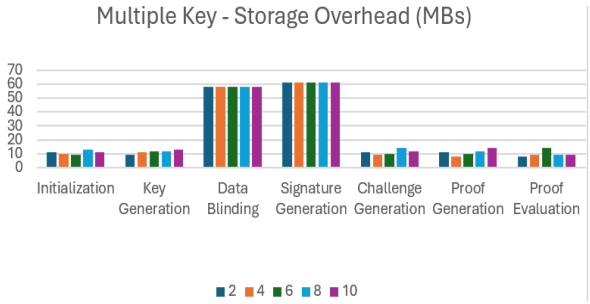

Fig. 5. Multiple key storage overhead.

D. Single Pair Implementation

Figure 6 demonstrates relatively lower storage overhead in comparison to Multiple Key Pair in phases like Initialization and Key Generation. The number of keys used affects how much storage is needed, especially during data binding and signature generation. The number of keys also affects communication overhead, especially during signature generation and data binding. Compressing encrypted data could help reduce storage requirements. When using multiple keys, storage demands increase significantly in data binding and signature generation phases. This increase is consistent across different numbers of Data Objects.

Some phases, like "Challenge Po" and "Response Po," have relatively stable storage needs. Communication overhead generally increases with the number of Data Objects. Certain phases like signature generation and data binding show fluctuations in communication overhead. The "Single Key Pair" implement consistently has lower communication

overhead in initial phases compared to "Multiple Key Pair".

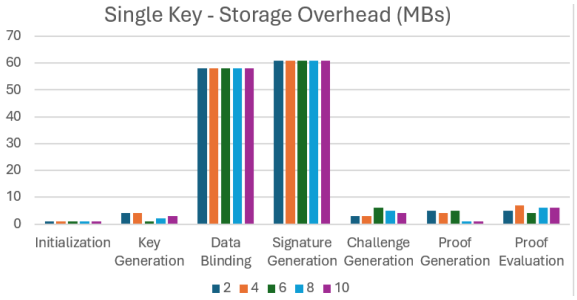

Fig. 6. Single key storage overhead.

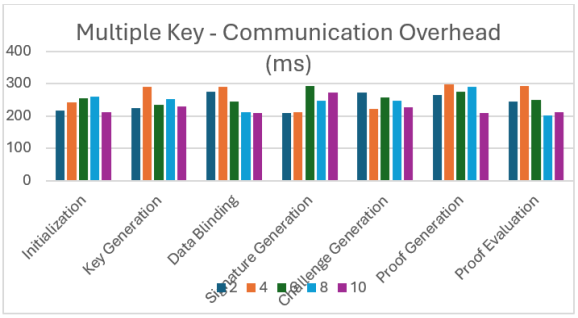

Fig. 7. Multiple key communication overhead.

E. Compression Techniques

Explore compression techniques for encrypted data to reduce storage requirements. Identify phases with significant storage and communication demands to optimize performance. Analyze communication patterns to streamline data exchange and reduce overhead. The findings highlight the trade-off between security and performance when using multiple keys. While it provides stronger security, it comes with increased storage and communication overhead shown in Fig. 7.

The research aimed to compare two methods for ensuring data integrity in cloud storage: using a single key pair and using multiple key pairs. Both methods have communication overhead, which increases with the number of Data Owners. The single key pair method generally has lower communication overhead in the initial phases. The multiple key pair method has higher communication overhead, especially in signature generation and key-related phases. The single key pair method is more efficient in initial phases like initialization and key generation. The multiple key pair method has increased computational demands, especially in signature generation and key-related phases. Both methods have varying storage requirements based on the number of data owners. The multiple key pair method consistently requires more storage, possibly due to managing multiple key pairs.

V. CONCLUSION

The research concludes that both methods have advantages and disadvantages. The single key pair method offers potential efficiencies in certain operational aspects, while the multiple key pair method introduces complexities impacting computational, communication, and storage overheads. The researcher suggests exploring advanced compression techniques to reduce communication overhead, optimizing storage utilization, and refining operational methodologies for both methods. Additionally, further research could delve into security considerations, scalability challenges, and real-time application viability for these implementations in various cloud environments. The research compared two ways to keep data safe in the cloud. One way is simpler but less efficient, while the other is more complex but potentially more secure. The researchers found that both methods have trade-offs in terms of communication, computation, and storage costs. They suggest that future research should focus on improving these methods to make them more efficient and secure.

ACKNOWLEDGEMENT

I wish to thank very much the people, who have helped in ensuring that this research is completed successfully. I would also like to express my gratitude to Mr. Ahthasham Sajid whose knowledgeable directions, open comments and consistent encouragement helped me through the process of this work. He has made exceptional contributions in terms of value and focus of this work as a teacher. I would also wish to acknowledge the good technical input and team spirit of Mr. Adnan Saleem my other team member. His wise talks, especially on cloud architecture and cryptographic mechanisms contributed greatly to the clarification of the practical part of the proposed auditing scheme. Lastly, I am especially grateful to Riphah International University for providing a supportive academic environment and the resources necessary to pursue this research. The university's commitment to excellence in education and research has played a vital role in my academic growth.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

The overall research effort was spearheaded by Muhammad Usman starting with the research problem identification and development of the study objectives. He formulated the fundamental set of the enhanced authenticator-based public auditing scheme, designed the algorithmic model, and did formal and experimental analysis to assess the performance of the systems. He made the first draft of the manuscript preparation, data interpretation organization and combined the results of the literature analysis and the suggested model outcomes.

Maxwell Soita assisted in the perfection of the research methodology and confirmation of the theoretical elements of the offered scheme. He helped to compare the computational and storage overheads of single and multiple key-pair implementations, is

making sure that the analytical results were accurate and consistent. Moreover, he gave feedback on the paper structure and technical flow, which is critical and made the findings to match the current standards of cloud security research.

Mohamed Ahmed was in favor of the experimental design and implementation stages especially on the cryptographic and auditing processes of the data integrity. He was involved in the validation of the results and visualization and he interpreted the data patterns concerning the storage and communication overheads. Mohamed helped in revising the manuscript too and provided considerable information on how to optimize security parameters in the cloud environment as well as increasing the clarity of the technical discussion.

CONFLICT OF INTERESTS

No conflict of interests was disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

REFERENCES

[1] Cloud Security Alliance, "Treacherous 12: Cloud Computing Top Threats in 2016," *CSA Official Press Release*, in press.

[2] K. Wu, Y. Li, L. Chen and Z. Wang, "Research of Integrity and Authentication in OPC UA Communication Using Whirlpool Hash Function," *Appl. Sci.*, vol. 5, no. 3, pp. 446–458, 2015.

[3] Y. Yu, 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. Informat. Forens. and Secur.*, vol. 12, no. 4, pp. 767-778, 2017.

[4] L. Jin, K. Li, Z. Li, F. Xiao, G. J. Qi and J. Tang, "Deep Semantic-Preserving Ordinal Hashing for Cross-Modal Similarity Search," *IEEE Trans. Neur. Netw. and Learn. Syst.*, vol. 30, no. 5, pp. 1429-1440, 2019.

[5] H. Li, L. Liu, C. Lan, C. Wang and H. Guo, "Lattice-based Privacy-preserving and Forward-secure Cloud Storage Public Auditing Scheme," *IEEE Access*, vol. 8, pp. 86797-86809, 2020.

[6] P. Singh and S. K. Saroj, "A Secure Data Dynamics and Public Auditing Scheme for Cloud Storage," in *2020 6th Int. Conf. Adv. Comput. and Commun. Syst.*, Coimbatore, India, pp. 695-700, 2020.

[7] H. Shacham and B. Waters, "Compact Proofs of Retrievability, Compact Proofs of Retrievability," *J. Ccryptol.*, vol. 26*,* no. 3, pp. 442-483, 2013.

[8] L. Rao, H. Zhang and T. Tu, "Dynamic Outsourced Auditing Services for Cloud Storage Based on Batch-Leaves-Authenticated Merkle Hash Tree," *IEEE Trans. Servic. Comput.*, vol. 13, no. 3, pp. 451-463, 2020.

[9] J. Yu, K. Ren and C. Wang, "Enabling Cloud Storage Auditing with Verifiable Outsourcing of Key Updates," *IEEE Trans. Informat. Forens. and Secur.*, vol. 11, no. 6, pp. 1362-1375, 2016.

[10] H. Wang, Q. Wu, B. Qin and J. Domingo-Ferrer, "Identity-based Remote Data Possession Checking in Public Clouds," *IET Informat. Secur.*, vol. 8, no. 2, pp. 114- 121, 2014.

[11] C. Wang, S. S. M. Chow, Q. Wang, K. Ren and W. Lou, "Privacy-preserving Public Auditing for Secure Cloud Storage," *IEEE Trans. Computers*, vol. 62, no. 2, pp. 362-375, 2013.

[12] H. Wang, D. He and S. Tang, "Identity-based Proxy-oriented Data Uploading and Remote Data Integrity Checking in Public Cloud," *IEEE Trans. Informat. Forens. and Secur.*, vol. 11, no. 6, pp. 1165-1176, 2016.

[13] H. Wang, D. He, J. Yu and Z. Wang, "Incentive and Unconditionally Anonymous Identity-based Public Provable Data Possession," *IEEE Trans. Servic. Comput.*, vol. 12, no. 5, pp. 824-835, 2019.

[14] B. Wang, B. Li and H. Li, "Knox: Privacy-preserving Auditing for Shared Data with Large Groups in the Cloud," in *Int. Conf. Appl. Cryptogr. and Netw. Secur.*, Berlin, Heidelberg, pp. 507-525, 2012.

[15] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu and R. Hao. "Enabling Public Auditing for Shared Data in Cloud Storage Supporting Identity Privacy and Traceability," *J. Syst. and Softw.*, vol. 113, pp. 130-139, 2016.

[16] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu and R. Hao, "Light-weight and Privacy Preserving Secure Cloud Auditing Scheme for Group Users via the Third Party Medium," *J. Netw. and Comput. Appl.*, vol. 82, pp. 56-64, 2017.

[17] P. Singh and S. K. Saroj, "A Secure Data Dynamics and Public Auditing Scheme for Cloud Storage," in *2020 6th Int. Conf. Adv. Comput. and Commun. Syst.*, Coimbatore, India, pp. 695-700, 2020.

[18] T. T. Do, K. Le, T. Hoang, H. Le, T. V. Nguyen and N. M. Cheung, "Simultaneous Feature Aggregating and Hashing for Compact Binary Code Learning," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4954-4969, 2019.

[19] A. Vamshi, G. J. Rao, S. K. Pasupuleti and R. Eswari, "EPF-CLPA: An Efficient Pairing-free Certificateless Public Auditing for Cloud-based CPS," in *2021 5th Int. Conf. Intellig. Comput. and Contr. Syst.*, Madurai, India, pp. 48-54, 2021.

[20] R. Zhou, M. He and Z. Chen, "Certificateless Public Auditing Scheme with Data Privacy Preserving for Cloud Storage," in *2021 IEEE 6th Int. Conf. Cloud Comput. and Big Data Analyt.*, Chengdu, China, pp. 675-682, 2021.

[21] H. Yan and W. Gui, "Efficient Identity-based Public Integrity Auditing of Shared Data in Cloud Storage with User Privacy Preserving," *IEEE Access*, vol. 9, pp. 45822-45831, 2021.

[22] Yogita and N. Kumar Gupta, "Integrity Auditing with Attribute Based ECMRSA Algorithm for Cloud Data Outsourcing," in *2020 3rd Int. Conf. Intellig. Sustain. Syst.*, Thoothukudi, India, pp. 1284-1289, 2020.

[23] X. Fan, F. Zhang, E. Turamat, C. Tong, J. H. Wu and K. Wang, "Provenance-based Classification Policy Based on Encrypted Search," in *2020 2nd Int. Conf. Industr. Artif. Intellig.*, Shenyang, China, pp. 1-6, 2020.

[24] R. V. Mante and N. R. Bajad, "A Study of Searchable and Auditable Attribute Based Encryption in Cloud," in *2020 5th Int. Conf. Commun. and Electron. Syst.*, Coimbatore, India, pp. 1411-1415, 2020.

[25] G. Ateniese, R. Di Pietro, L. V. Mancini and G. Tsudik, "Scalable and Efficient Provable Data Possession," in *Proc. 4th Int. Conf. Secur. and Privac. Commun. Netw.*, no. 9, pp. 1-10. 2008.

[26] F. Wang, L. Xu, K. K. R. Choo, Y. Zhang, H. Wang and J. Li, "Lightweight Certificate-based Public/Private Auditing Scheme Based on Bilinear Pairing for Cloud Storage," *IEEE Access*, vol. 8, pp. 2258-2271, 2020.

[27] X. Gao, J. Yu, Y. Chang, H. Wang and J. Fan, "Checking Only When It Is Necessary: Enabling Integrity Auditing Based on the Keyword with Sensitive Information Privacy for Encrypted Cloud Data," *IEEE Trans. Depend. and Secur. Comput.*, vol. 19, no. 6, pp. 3774-3789, 2022.

[28] B. Wang, B. Li and H. Li, "Oruta: Privacy-preserving Public Auditing for Shared Data in The Cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 43-56, 2014.

[29] H. Shacham and B. Waters, "Compact Proofs of Retrievability, Compact Proofs of Retrievability," *J. Cryptol.*, vol. 26*,* no. 3, pp. 442-483, 2013.

[30] Z. Fu, X. Sun, Q. Liu, L. Zhou and J. Shu, "Achieving Efficient Cloud Search Services: Multi Keyword Ranked Search Over Encrypted Cloud Data Supporting Parallel Computing," *IEICE Trans. Commun.*, vol. 98, no. 1, pp. 190–200, 2015.

[31] W. Shen, J. Qin, J. Yu, R. Hao and J. Hu, "Enabling Identity-Based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage," *IEEE Trans. Informat. Forens. and Secur.*, vol. 14, no. 2, pp. 331-346, 2019.

[32] Y. Luo, M. Xu, S. Fu, D. Wang and J. Deng, "Efficient Integrity Auditing for Shared Data in the Cloud with Secure User Revocation," in *2015 IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, pp. 434-442, 2015.

[33] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as The 5th Utility," *Future Generat. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[34] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[35] Z. Xia, X. Wang, X. Sun and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Trans. Parall. and Distrib. Syst.*, vol. 27, no. 2, pp. 340-352, 2016.

[36] J. Shen, H. Tan, S. Moh, I. Chung, Q. Liu and X. Sun, "Enhanced Secure Sensor Association and Key Management in Wireless Body Area Networks," *J. Commun. and Netw.*, vol. 17, no. 5, pp. 453-462, 2015.

[37] M. K. Srinivasan, K. Sarukesi, P. Rodrigues, M. S. Manoj and P. Revathy, "State of-The Art Cloud Computing Security Taxonomies: A Classicizing of Security Challenges in the Present Cloud Computing Environment," in *Proc. Int. Conf. Adv. Comput., Commun. and Informat.*, pp. 470-476, 2012.

[38] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *Nat. Inst. Stand. and Technol. Spec. Public.,* vol. 53, pp. 1-7, 2011.

[39] F. Wang, L. Xu and W. Gao, "Comments on "SCLPV: Secure Certificateless Public Verification for Cloud-Based Cyber-Physical-Social Systems Against Malicious Auditors," *IEEE Trans. Computat. Soc. Syst.*, vol. 5, no. 3, pp. 854-857, 2018.

[40] J. Sun and Y. Fang, "Cross-Domain Data Sharing in Distributed Electronic Health Record Systems," *IEEE Trans. Parall. and Distrib. Syst.*, vol. 21, no. 6, pp. 754-764, 2010.