Integrated Privacy-Preserving Auditing: Synthesizing Cloud, Edge, and IoT Integrity Mechanisms

Andrew Photinakis
acp7795@rit.edu
Rochester Institute of Technology
Rochester, New York, USA

Abstract

Distributed computing environments encompasses cloud, edge, and Internet of Things (IoT) infrastructures. The rise in these environments has created a critical need for robust data integrity verification. Traditional auditing mechanisms, while effective in centralized settings, frequently fail in heterogeneous systems. The reason is due to excessive metadata leakage, reliance on full data access, or prohibitive computational overhead. Such limitations expose sensitive operational patterns in regulated sectors such as healthcare and industrial automation, where privacy and scalability are paramount. This research synthesized and evaluated five contemporary auditing frameworks to determine their efficacy in addressing challenges within a unified, privacy-preserving architecture.

The analysis revealed that by integrating time-based auditing, homomorphic signatures, and identity-based cryptography, tradeoffs between security and performance are significantly mitigated. The evaluation of hardware-assisted techniques, such as Polynomial Prefix Message Authentication Codes combined with Physical Unclonable Functions, demonstrated a reduction in computational overhead by up to 3 times and a decrease in communication costs by nearly one-third compared to static methods. The study highlighted that dynamic data support is essential for compliance with modern privacy regulations. This concludes that a layered security model, merging edge-driven access control with cryptographic verification, provides the necessary end-to-end trust for scalable, regulated distributed systems.

Keywords

Cloud computing, edge computing, Internet of Things (IoT), data integrity, privacy-preserving auditing, identity-based cryptography, time-based auditing, access control

1 Overview

Distributed computing systems, which're are growing fast, present significant challenges for maintaining data security and privacy [1]. These setups handle vast amounts of sensitive information from factory machines, smart sensors, and edge devices, often outsourcing storage and processing to third-party servers. Traditional auditing methods frequently fail in heterogeneous systems. Wang et al. [2] demonstrate that the methods often expose sensitive metadata. Ullah et al. [3] note that they can introduce significant computational overhead that creates bottlenecks. Gai et al. [4] emphasize that reliance on full data access during transfer operations is inefficient for large datasets. With global data generation expected to double rapidly, developing efficient, privacy-preserving auditing solutions is critical.

Ensuring data integrity while preserving privacy is critical, especially in sectors with high real-world importance. Integrity checks are essential to detect unauthorized changes, accidental corruption, or deliberate deletions. While privacy-focused auditing prevents leakage of sensitive metadata such as file identifiers or access patterns. Beyond requirements, modern systems must also support dynamic data operations, allowing users to update or delete records without breaking the audit chain [3]. In unmonitored environments, the physical security of edge devices becomes a prerequisite for trust, protecting against attacks that could compromise keys stored in memory [1].

Despite the critical nature of protections, current research addresses these challenges in isolation. Existing frameworks typically focus either on cloud-centric scalability or edge-centric access control, rarely bridging the gap between them. For instance, some protocols are optimized for dynamic data updates, they neglect the physical security vulnerabilities in IIoT devices. Hardware-assisted solutions may not scale effectively to handle the massive data volumes of a centralized cloud. This disconnect highlights the need for a comprehensive review of disparate auditing mechanisms to understand how they might be synthesized into a unified, privacy-preserving architecture.

The remainder of this paper is organized as follows. Section 2 reviews related work in the field of privacy-preserving auditing across cloud, IoT, and edge environments. Section 3 defines the specific research problem, highlighting the challenges of heterogeneous systems and the limitations of existing approaches. Section 4 presents a detailed comparative analysis of five foundational frameworks, evaluating them against privacy, efficiency, and scalability criteria. Section 5 and Section 6 discuss the legal and ethical considerations of these technologies, respectively. Section 7 concludes the report and suggests directions for future research.

2 Related Work

Privacy-preserving auditing and data integrity checks have evolved across all types of distributed computing environments. This evolution spans from foundational cryptographic proofs to modern, edge-aware architectures that address dynamic data and physical security.

2.1 Foundational Auditing Frameworks

Remote data integrity is prevelant in two foundational paradigms: Provable Data Possession (PDP) and Proofs of Retrievability (POR). Ateniese et al. [5] first introduced PDP, a scheme allowing a client to verify that a server possesses the original data without retrieving it. It relies on RSA-based homomorphic tags for blockless verification. Concurrently, Juels et al. [6] propose POR, which utilize sentinels embedded in the data to detect corruption and ensure data recoverability. Shacham et al. [7] refine these concepts with compact proofs based on BLS signatures, which significantly reduced communication overhead. Boneh et al. [8] provide underlying short signature schemes that made compact proofs feasible. Early frameworks established the mathematical basis for remote integrity checks, primarily addressing static data in centralized servers. However, they lacked the dynamic and privacy-preserving features required for modern IoT environments.

2.2 Cloud Auditing and Privacy

Building on foundational schemes, recent research in cloud auditing has focused on metadata privacy and temporal verification. Wang et al. [2] proposed a time-based auditing approach that utilizes metadata-hiding tags to conceal file details and counts, facilitating secure batch and period-based checks. The approach leverages identity-based cryptography to eliminate certificate management overhead and employs virtual files to mitigate guessing attacks. Other works have extended PDP and POR for multi-user cloud environments, shifting the focus from basic integrity checks to privacy-centric systems, although challenges in handling dynamic data persist [9, 10].

Identity-based auditing and secure data transfer protocols aim to reduce computational load while preserving privacy during ownership changes. Gai et al. [4] developed the Privacy-Preserving Identity-Based Public Auditing (PPADT) scheme, which uses pseudonyms, grouped verifiers, and random masking to achieve constant-time data transfer costs, significantly reducing the overhead associated with traditional linear methods. Foundational research in identity-based cryptography has established that such systems can simplify key management and protect privacy, although they often lack specific optimizations for edge environments [11], [12].

2.3 Dynamic IoT and Edge Security

Dynamic IoT auditing addresses the need for frequent updates, resource constraints, and automated verification. Ullah et al. [3] introduced Privacy-Aware Secure Data Auditing (PASDA) framework, which employs homomorphic signatures and self-triggering mechanisms to automatically verify data integrity. The system supports multiple file versions and manages user termination, which builds on established signature schemes that're optimized for high-speed, limited-bandwidth IoT data streams.

Cryptographic protocols in IIoT and edge computing are increasingly integrated with hardware security features to enable reliable auditing. Shan et al. [1] combine Polynomial Prefix Message Authentication Codes (P2MAC) with Physical Unclonable Functions (PUFs). This was to enable secure verification and selective data access. Rahman et al. [13] extend the combination of P2MACs and PUFs by integrating Edge Data Integrity Verification (EDIV) directly with access control mechanisms. This enables enforcement of real-time security policies at the edge. By leveraging computing principles and ABAC models, a shift is presented from purely cloud-based audits to end-to-end security architectures that combine physical and digital defenses [14], [15].

2.4 Emerging Paradigms: Deduplication, Searchability, and Decentralization

Research beyond basic integrity verification increasingly examined storage efficiency as cloud infrastructures scaled. Gao et al. [16] propose an auditing scheme that supported deduplication to mitigate redundancy. By ensuring that the cloud stored only a single instance of duplicate files, this design reduced the volume of stored authenticators, minimizing storage overhead [16].

Research on storage auditing used decentralized mechanisms to reduce dependence on a centralized Third-Party Auditor (TPA). Du et al. [17] applied blockchain consensus to replicate audit logs across multiple verifiers. This design removed the single point-of-failure risk associated with TPA-based models. Tsigkanos et al. [18] introduced runtime verification protocols for IoT systems, enabling continuous evaluation of properties. This approach addresses the limitations of such frameworks that operate only at fixed or statically defined intervals.

Collectively, the studies illustrate the progression from basic cloud audits to dynamic, edge-aware frameworks, highlighting the ongoing need for fully integrated solutions across the cloud-edge-IoT continuum.

3 Research Problem

With the rise of diverse distributed systems presents a critical research challenge. That is ensuring data integrity and confidentiality across high-velocity, resource-constrained, and heterogeneous environments. As mentioned in Section 1, legacy auditing mechanisms fail to scale in these complex scenarios. These methods are constrained by dependencies on full data access and high computational costs. Due to this, applicability is limited in modern architectures where data flows and updates occur continuously [2], [3], [4]. With global data volumes projected to double every two years, affecting over 70% of industries by 2025, developing efficient, privacy-preserving auditing solutions is imperative.

3.1 Challenges in Heterogeneous Systems

Auditing in heterogeneous environments accumulates by the sheer diversity of distributed systems. This complicates the deployment of unified auditing approaches and results in inconsistent security coverage. Resource-limited IoT and edge devices face heavy computational and data transfer demands that can severely degrade performance if traditional, heavyweight auditing protocols are applied [3]. Existing tools often lack mechanisms to handle dynamic operations efficiently, struggling to support block-level updates, deletions, and version management without incurring prohibitive overhead [4].

Beyond operational constraints, ensuring security and privacy remains difficult. The issues are that sensitive metadata, file identifiers, and access patterns are vulnerable to exposure, allowing unauthorized Third-Party Auditors (TPAs) to infer proprietary operational habits [2]. Physical security for edge and IIoT devices is also frequently overlooked. Shan et al. [1] note that devices in unattended environments are particularly susceptible to physical tampering and side-channel attacks. Rahman et al. [13] emphasize that these vulnerabilities can compromise the reliability of data at the edge, which undermines the trust required for later access control decisions.

3.2 Limitations of Existing Approaches

While prior studies address components of the challenges, solutions do not suffice for modern heterogeneous environments. Although Wang et al. [2] utilize tags to conceal metadata in cloud audits, their reliance on virtual files struggles with dynamic changes and seamless edge integration. Similarly, while Ullah et al. [3] enable automated integrity checks for dynamic IoT data, their architecture depends on trusted intermediary hubs, which introduces a centralized bottleneck. Gai et al. [4] address scalability via constant-time data transfers, yet their identity-based scheme leaves specific edge and hardware security requirements unaddressed. Shan et al. [1] provide robust hardware-rooted security through PUFs, but their focus on static prefixes offers limited support for dynamic content updates. Finally, Rahman et al. [13] enforce edge policies through EDIV and ABAC, but primarily address access control rather than underlying cryptographic efficiency.

The core analytic contribution of this work is the synthesis of techniques into a unified conceptual framework. By combining metadata-hiding, time-based auditing with hardware-rooted security and edge-driven access control, the proposed framework addresses integrity, privacy, efficiency, dynamic content support, and physical security across the cloud-to-edge-IoT continuum. This integrated approach fills a critical gap, leveraging the strengths of existing methods to comprehensively address the challenges inherent in modern distributed systems.

4 Comparative Analysis

The purpose of this comparative analysis is to systematically evaluate the five foundational studies informing this report, identifying both the common design philosophies and the critical technical divergences within the domain of distributed data auditing. Current literature often addresses the challenges of data integrity verification (DIV) and access control (AC) in isolation. This section establishes the basis for a holistic evaluation to determine which architectural choices-spanning from hardware integration to algorithmic efficiency are best suited for a heterogeneous Cloud-to-Edge IoT ecosystem.

The evaluation criteria for this analysis focus on several key axes critical for secure, scalable deployment in real-world, resource-constrained, and regulated environments.

- **Privacy level and scope** measure the depth of protection, particularly the ability to conceal sensitive metadata such as file counts, identities, and timestamps, as well as user identities through pseudonyms, typically from an semi-honest Third-Party Auditor (TPA) [4], [3], [1].
- **Efficiency and scalability** assess the computational overhead on constrained devices and servers, emphasizing operations with $O(1)$ efficiency and empirical verification times in milliseconds [4], [1], [2].
- **Applicability and focus** consider the primary infrastructure layer targeted—Cloud, IoT/IIoT, or Edge—and the nature of the data flow, including Cloud-to-Edge verification, Edge-to-End access control, or centralized cloud auditing [13], [2].
- **Dynamic data support** evaluates the ability to handle non-static information, from basic block-level modifications such as insertions, updates, and deletions to complex operations like verifiable data ownership transfer or managing multiple file versions [4], [3].
- **Trust model and security root** compares foundational mechanisms, ranging from reliance on pure cryptographic hardness, such as bilinear pairings, to specialized hardware-based solutions, including Physical Unclonable Functions (PUFs) and Trusted Execution Environments (TEEs), which address physical security threats [1], [13].

4.1 Common Themes

Throughout the selected research, a comparative analysis reveals important and recurring themes. They highlight shared challenges and converging solutions in the field of privacy-preserving data auditing for cloud-to-edge systems.

All frameworks prioritize **privacy-preserving auditing** to prevent TPAs from inferring sensitive data. While implementation varies—ranging from random masking in Gai et al. [4] to ABAC in Rahman et al. [13] share objective is to decouple integrity verification from data visibility. This is frequently paired with **Identity-Based Cryptography (IBC)** or pseudonymity to simplify the complex certificate management inherent in traditional PKI, protecting user identities from auditors [2].

To ensure scalability, the surveyed works adopted **homomorphic authenticators** (such as BLS signatures or Homomorphic Hash Functions). These allow multiple data blocks to be aggregated into a single, compact proof. The result is enabling verification costs that are effectively independent of the data size ($O(1)$) [1], [3]. Cryptographic efficiencies are critical for handling the massive data volumes characteristic of modern distributed systems without imposing prohibitive bandwidth costs.

There exists a consensus that modern frameworks must support **data dynamics**, ensuring that security does not come at the cost of operational flexibility. Whether through block-level updates for high-velocity IoT streams [3] or secure ownership transfers for corporate assets [4], the ability to modify or transfer data without breaking the audit chain is a fundamental requirement. Minimizing computational overhead remains paramount, with recent hardware-assisted designs demonstrating significant empirical speedups over traditional static methods [1].

4.2 Divergent Themes

While the selected papers share common goals, their approaches diverge significantly based on their target system, threat model, and primary functional objectives. These divergences highlight the specialized, non-monolithic nature of modern data auditing.

4.2.1 System Focus (Cloud vs. IoT/IIoT vs. Edge).

The target environment shapes resource assumptions, adversarial capabilities, and system objectives. Cloud-centric designs [4], [2] assume abundant storage and computation, focusing on scaling to large centralized datasets while preserving privacy against powerful, untrusted cloud providers or TPAs. IoT and IIoT systems [3], [1] operate under constrained resources, emphasizing minimal computational burden on end devices. IIoT deployments additionally contend with limited physical protection, as devices may be exposed to tampering. Edge-centric auditing [13] mediates heterogeneous data flows between cloud and devices, performing both cloud-to-edge verification and edge-to-device access control. Each orientation entails trade-offs: cloud methods provide scalability but are ill-suited for resource- or physically-constrained environments, IIoT architectures offer strong security guarantees yet depend on specialized hardware such as PUFs and TEEs, and edge-oriented frameworks deliver practical end-to-end enforcement while functioning primarily as architectural rather than purely cryptographic solutions.

4.2.2 Hardware-Assisted vs. Pure Cryptography.

The surveyed work distinguishes between pure cryptographic and hardware-assisted designs based on trust in the physical security of the device. Pure cryptographic systems assume device memory remains secure, storing private keys in memory and relying on the hardness of problems such as the Discrete Logarithm and Computational Diffie–Hellman, with a threat model that excludes physical compromise [4], [2]. Hardware-assisted designs treat device memory as insecure, regenerating keys on demand through a Physical Unclonable Function that derives secrets from unique micro-structural variations hidden within the device's physical substrate, often supplemented by a Trusted Execution Environment to isolate and protect audit computations [1]. Shan et al. [1] present the only hardware-assisted design, while all other works rely solely on conventional cryptographic primitives, including bilinear pairings [2], [4], homomorphic verifiable authenticators [3], or symmetric and hash-based enforcement [13]. Pure cryptography prioritizes portability and broad deployment on commodity hardware, whereas hardware-assisted approaches address a stronger threat model in unattended IIoT environments, trading wider deployability for greater resilience against physical compromise [1].

4.2.3 Scope of Dynamic Data Support.

Data dynamics vary across the surveyed literature, reflecting differences in the operations supported. Content dynamics enable block-level insertion, update, and deletion, requiring data structures that update authenticators efficiently while preserving verifiability. Ullah et al. [3] implement this through a linked-list mechanism with a timestamp with a pointer, maintaining multiple file versions. Ownership dynamics operate at the file level, focusing on reassigning ownership without recomputing authenticators, as in Gai et al. [4]. Static or snapshot auditing treats data as immutable, verifying it at a specific time or upon access, exemplified by Wang et al. [2], Rahman et al. [13], and Shan et al. [1], where files remain static despite flexible access.

Among the surveyed systems, Ullah et al. [3] uniquely supports full content dynamics, providing explicit algorithms for insertion, update, and deletion via a version-linked design. Ownership transfer, referred to as "efficient data transfer" in Gai et al. [4], is exclusive to that work and does not allow block-level modifications. Static snapshot models dominate traditional cloud auditing, including Wang et al. [2] with time-period-based verification and Rahman et al. [13] where auditing occurs upon end-device access. Shan et al. [1] allow flexible access patterns but audits over static files.

These differing scopes reflect deployment objectives. Full content dynamics [3] address continuously evolving IoT data streams, while ownership dynamics [4] support business scenarios requiring efficient reassignment of file control. The trade-offs are significant: Ullah et al. [3] incurs substantial state-management overhead, whereas Gai et al. [4] achieves constant-time ownership reassociation by excluding block-level modifications, making it specialized rather than a general-purpose dynamic auditing framework.

4.2.4 Auditing Trigger and Mechanism.

Systems differ in what initiates integrity verification, spanning user- or TPA-driven, event-driven, access-driven, and non-interactive approaches. In user- or TPA-initiated models, auditing is reactive as the auditor determines when verification occurs and sends a challenge, as in Gai et al. [4] and Wang et al. [2]. Event-driven auditing automatically detects changes and issues alerts, exemplified by Ullah et al. [3], reducing reliance on manual triggers. Access-driven verification integrates auditing into the data-access pipeline, as in Rahman et al. [13], ensuring data is verified before use. Shan et al. [1] implements a non-interactive mechanism within a Trusted Execution Environment, generating verification challenges internally and preventing network-level tampering.

These approaches involve trade-offs in timeliness, trust, and deployability. Event-driven auditing enables continuous monitoring but assumes the untrusted cloud honestly reports changes. Access-driven verification guarantees integrity at the moment of use but may introduce latency. Non-interactive auditing prevents challenge forgery but requires TEE hardware, limiting deployment. User- or TPA-initiated auditing is simple and widely deployable yet offers no assurance between audit events and cannot guarantee verification prior to data consumption.

4.2.5 Primary Privacy Target.

Although all surveyed systems aim to preserve privacy, they differ in the adversary they primarily protect against, which shapes their technical design. In TPA-focused models, the goal is to hide data content and metadata from an honest-but-curious auditor during verification [4], [2]. Wang et al. [2] blind the auditor to file counts and identities, while Gai et al. [4] additionally apply random masking and pseudonyms to conceal both data content and ownership.

A distinct approach targets the data consumer. Privacy-preserving mechanisms in this model allow an authenticated user to access only a portion of a shared file while cryptographically preventing access to the remainder [1]. Shan et al. [1] exemplify this strategy, enabling flexible verification of data prefixes for legitimate consumers rather than auditors.

Some systems extend protection to all external parties, including the cloud server and the auditor. In Ullah et al. [3], the user blinds data before submission to a cluster head, which re-blinds it prior to storage or processing, preventing both the server and auditor from inferring the user's identity or data content. Different designs reflect distinct motivations, with TPA-focused systems proving possession without revealing data, consumer-focused models enabling selective sharing in IIoT environments, and all-party protection maximizing confidentiality. Each approach is specialized, as TPA-focused solutions do not protect consumer privacy and consumer-focused designs do not safeguard auditor privacy, showing that technical choices correspond directly to specific privacy threats.

4.3 Additional Analysis

The surveyed works differ in how clearly they present their contributions, the completeness of their technical solutions, and the significance of their research. Wang et al. [2] achieve clarity through a precise problem definition that focuses on time-based auditing while concealing metadata. Auxiliary tags hide file identities and virtual files mask file counts, with formal proofs of correctness and privacy preserving directly supporting the design goals. Ullah et al. [3] provide a structured presentation with nine algorithms and diagrams that clarify the roles of the Key Generation Center (KGC), Third-Party Cloud Server (TPCS), and Cluster Head (CH), particularly for automated self-triggering and workflows for insertion, updates, and deletions. Gai et al. [4] maintain focus by identifying inefficiencies in linear-cost data transfers and presenting a constant-time solution, with system workflows illustrating the flow of transformation values between Previous Owner (PO), Current Owner (CO), and Third-Party Auditor (TPA). Shan et al. [1] simplify a complex hardware-software co-design through diagrams that span PUF-based key generation to TEE challenges, with a thorough threat model addressing physical attacks. Rahman et al. [13] emphasize architectural clarity by demonstrating how Edge Data Integrity Verification (EDIV) feeds into the ABAC engine, addressing research gaps where integrity and access control were previously isolated.

Regarding completeness, Ullah et al. [3] provide the most extensive algorithmic coverage, supporting block-level content dynamics, multi-versioning, and user revocation, though this results in complexity. Shan et al. [1] achieves comprehensive system-level security by integrating physical device protection, secure computation, and cryptographic verification, with evaluation demonstrating real-world speedups. Gai et al. [4] offers a specialized contribution limited to efficient ownership transfer, highly deployable for specific business scenarios but not general-purpose. Wang et al. [2] focuses on time-based metadata hiding without addressing dynamic content. Rahman et al. [13] provides an architectural framework integrating EDIV and ABAC without introducing new cryptographic protocols.

The significance of each work varies by focus. Wang et al. [2] shift auditing from file-centric to time-centric, supporting users who think in time periods. Gai et al. [4] addresses scalability in ownership transfer, enabling practical movement of massive datasets. Ullah et al. [3] advances block-level dynamics and introduces automated self-triggering, moving auditing from a passive to an active security model. Shan et al. [1] bridges cryptography and hardware security, providing robust protection against physical compromise. Rahman et al. [13] unifies integrity verification and access control, treating integrity as a prerequisite for data access.

These observations reveal trends guiding analysis. Auditing is moving from monolithic, cloud-centric approaches toward specialized, layered architectures that accommodate IoT resource constraints [3], IIoT physical risks [1], and the gatekeeper role of the Edge [13]. Privacy has evolved from protecting content and identity [4] to concealing metadata [2] and supporting selective disclosure [1]. Performance bottlenecks differ by protocol, with Gai et al. [4] addressing transfer scalability, Shan et al. [1] optimizing device computation, and Ullah et al. [3] managing dynamic updates efficiently. These factors demonstrate that a single auditing model cannot satisfy all use cases; specialized solutions aligned with deployment context, threat models, trust assumptions, and resource constraints are essential.

5 Legal Considerations

Privacy-preserving auditing systems operate within a strict framework of data privacy and security regulations. Compliance requires aligning technical architectural choices with specific legal mandates regarding integrity, minimization, and accountability.

5.1 Regulatory Overview

The General Data Protection Regulation (GDPR) establishes global standards for auditing requirements. Article 5(1)(f), *Integrity and Confidentiality*, mandates secure processing to prevent loss or destruction, providing the legal basis for data integrity auditing. Article 5(1)(c), *Data Minimisation*, restricts data collection to necessary elements, applying equally to content and metadata. Article 5(1)(d), *Accuracy*, and Article 32, *Security of Processing*, obligate controllers to rectify inaccuracies and implement measures ensuring the continuous integrity of processing systems.

In the United States, the California Consumer Privacy Act (CCPA) grants consumers rights to correct inaccurate information and mandates reasonable security procedures. Concurrently, the proposed EU Cyber Resilience Act (CRA) targets IoT and IIoT infrastructure, enforcing *security-by-design* principles and assigning liability to manufacturers for vulnerabilities. The NIST Privacy Framework further structures these obligations through its *Protect* and *Govern* functions, which align with robust auditing mechanisms.

5.2 Impact on Privacy-Preserving Auditing

The GDPR data minimization principle transforms metadata privacy from a technical feature into a legal necessity. Protocols exposing unnecessary metadata, such as file identities or access patterns, risk non-compliance. The GDPR and CCPA rights regarding data rectification render static-only auditing systems insufficient; compliant systems must support verifiable data updates and modifications.

For IoT and IIoT environments, the EU CRA places liability on manufacturers for device security. Device-level integrity mechanisms, including Physical Unclonable Functions (PUFs), become essential for managing legal risk. Simultaneously, GDPR Articles 5(1)(f) and 32 elevate auditing from a best practice to a mandatory compliance activity, requiring organizations to maintain ongoing verification of data integrity.

5.3 Regulatory Alignment of Surveyed Frameworks

Wang et al. [2] implement GDPR Art. 5(1)(c) by concealing metadata unnecessary for integrity checks, utilizing a virtual file mechanism to ensure compliance during idle periods. Ullah et al. [3] facilitate compliance with the *Right to Rectification* (GDPR Art. 5(1)(d)) and the *Right to Correct* (CCPA) by providing full content dynamics via the `Updationprocess2CS` algorithm.

Gai et al. [4] address data portability and ownership transfer with an efficient $O(1)$ mechanism. Shan et al. [1] align with the EU CRA's mandate for security-by-design by enforcing hardware-level protections with PUFs, safeguarding IIoT devices from physical tampering. Rahman et al. [13] operationalize GDPR Art. 32 by integrating Edge Data Integrity Verification (EDIV) into Attribute-Based Access Control (ABAC), ensuring access is granted only upon proven integrity.

5.4 Organizational and Societal Implications

Non-compliance with regulations such as GDPR carries severe financial risks, including penalties up to 4% of global annual revenue. Beyond financial implications, failure to maintain auditable data integrity erodes trust; organizations unable to prove data integrity or those exposing sensitive metadata risk reputational damage. In corporate situations, such as company acquisitions described by Gai et al. [4], data is a core property. The inability to verify transfer of data ownership and prove integrity during transactions can devalue assets and obstruct business operations. Similarly, Ullah et al. [3] highlight that in sectors such as banking and tax administration, automated management systems rely on trusted data feeds. Undetected corruption in this systems risks not only audit failures but organizational reputational damage.

Integrity failures in critical infrastructure, including healthcare and industrial control systems, create substantial legal liability and physical safety risks. Shan et al. [1] warn that without robust physical protections, industry clouds are vulnerable to sabotage where configuration data is manipulated to disrupt entire systems, such as power grids or smart meters. In the public sector, Ullah et al. [3] record that honor breaches can corrupt electronic health records or criminal justice data, which can lead to life-threatening errors or legal setbacks. The frameworks proposed by Shan et al. [1] and Rahman et al. [13] mitigate these risks by ensuring data is verifiably trustworthy prior to utilization.

6 Ethical Considerations

Beyond legal compliance, the design of auditing systems carries significant ethical weight. The ACM Code of Ethics (2018) provides a framework for evaluating these responsibilities.

6.1 Relevant ACM Ethical Principles

ACM 1.2, Avoid Harm, obligates computing professionals to mitigate negative consequences. In auditing, harm can result from integrity failures (e.g., corrupted medical records), privacy breaches (an auditor learns sensitive metadata), or physical device compromise (an IIoT sensor is hacked). The research addresses these risks: Shan et al. [1] prevent physical device attacks using PUFs; Rahman et al. [13] block access to data failing integrity checks; and Wang et al. [2] and Gai et al. [4] protect against privacy breaches by blinding the TPA to metadata and user identities.

ACM 1.6, Respect Privacy, demands utmost care with personal information and a clear understanding of its provenance. In auditing, this applies to both content and metadata. The surveyed research exemplifies this principle: Wang et al. [2] safeguard metadata; Shan et al. [1] allow consumers to verify data prefixes without exposing private suffixes; and Gai et al. [4] preserve identity privacy with pseudonyms.

ACM 2.5, Comprehensive Evaluation, requires analysis of limitations and risks, ensuring transparency about capabilities and trust assumptions. The papers adhere to this by explicitly defining threat models: Gai et al. [4] and Wang et al. [2] describe their TPA as honest-but-curious; Shan et al. [1] include physical attacks; and Wang et al. [2] note the unavoidable overhead costs of privacy mechanisms.

6.2 Application and Broader Societal Responsibilities

Ethical auditing requires balancing transparency and obfuscation. The `virtual file` mechanism in Wang et al. [2] hides empty time periods ($n_t = 0$) from the TPA, protecting privacy while raising transparency concerns. This necessitates weighing the duty to protect privacy (ACM 1.6) against the duty to be honest and trustworthy (ACM 1.3). Ethical design necessitates anticipating misuse. By alerting users to integrity-violating changes, the `automated self-triggering` system in Ullah et al. [3] prevents harm (ACM 1.2). However, reporting all changes indiscriminately could enable user surveillance, underscoring the need for careful implementation.

Hardware-based solutions introduce broader societal considerations regarding equity. While the PUF and TEE mechanisms in Shan et al. [1] provide robust security for high-value IIoT infrastructure, they rely on specialized hardware, which may leave low-cost consumer IoT devices vulnerable. This creates a security divide, worsening digital inequalities (ACM 1.4) where only organizations that have a surplus of resources can afford top-tier integrity. To counter this, architectures used by Rahman et al. [13] promote privacy by default. This approach enforces integrity verification before granting access, which embeds security into logic rather than just hardware. This demonstrates a more inclusive and responsible system design.

7 Conclusion

The analysis of privacy-preserving auditing techniques indicates a shift from monolithic, cloud-centric solutions toward specialized, layered security architectures. Protocols must align with specific deployment constraints; high-scale cloud auditing requires $O(1)$ efficiency for data transfer [4], while Industrial IoT (IIoT) demands hardware-based security mechanisms (PUFs and TEEs) to mitigate physical attacks [1]. The definition of privacy now encompasses strict metadata protection [2] and selective disclosure to consumers [1]. Empirical results show up to a threefold speedup in computation and significant reductions in communication costs compared to conventional schemes [1].

End-to-end security requires proactive management of dynamic data rather than passive verification. Solutions address this requirement via algorithmic support for content dynamics [3] or the integration of integrity verification (EDIV) with access control (ABAC) at the edge [13]. These mechanisms establish the technical foundation for compliance with legal mandates such as GDPR and the EU Cyber Resilience Act, fulfilling ethical and regulatory obligations.

Future research must optimize the trade-offs inherent in these systems. Key directions include developing lightweight cryptographic primitives to reduce overhead in complex dynamism protocols [3] and standardizing hardware security models to lower IIoT adoption barriers [1]. Additionally, decentralized auditing frameworks leveraging blockchain technology offer potential for enhancing transparency and distributed trust across the cloud-to-edge pipeline.

References

[1] Xiaohu Shan, Haiyang Yu, Yurun Chen, Yuwen Chen, and Zhen Yang. S2a-p2fs: Secure storage auditing with privacy-preserving flexible data sharing in cloud-assisted industrial iot. *IEEE Transactions on Mobile Computing*, 24(7):5699–5715, 2025.

[2] Min Wang, Jia Yu, Wenting Shen, and Rong Hao. Privacy-preserving time-based auditing for secure cloud storage. *IEEE Transactions on Information Forensics and Security*, 19:7866–7878, 2024.

[3] Fasee Ullah, Chi-Man Pun, Muhammad Ismail Mohmand, Rakesh Kumar Mahendran, Arfat Ahmad Khan, Sarah M. Alhammad, Joel J. P. C. Rodrigues, and Ahmed Farouk. Privacy-aware secure data auditing for cloud-based intelligence of things environment. *IEEE Internet of Things Journal*, 12(11):15288–15303, 2025.

[4] Chao Gai, Wenting Shen, Ming Yang, and Jia Yu. Ppadt: Privacy-preserving identity-based public auditing with efficient data transfer for cloud-based iot data. *IEEE Internet of Things Journal*, 10(22):20065–20079, 2023.

[5] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 598–609, 2007.

[6] Ari Juels and Burton S. Kaliski. Pors: Proofs of retrievability for large files. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 584–597, 2007.

[7] Hovav Shacham and Brent Waters. Compact proofs of retrievability. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 90–107, 2008.

[8] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532, 2001.

[9] Bin Wang, Hui Li, and Ming Li. Privacy-preserving public auditing for dynamic group based on hierarchical tree. *Journal of Computer Research and Development*, 54(12):2657–2669, 2017.

[10] Chang Liu, Rajiv Ranjan, Chi Zhang, Caspar Yang, Dimitrios Georgakopoulos, and Jinjun Chen. Public auditing for big data storage in cloud computing – a survey. *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 1124–1132, 2013.

[11] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 47–53, 1984.

[12] Jung Hee Cha and Jae Hee Cheon. An identity based signature from gap diffie-hellman groups. In *International Workshop on Public Key Cryptography*, pages 18–30, 2003.

[13] Saifur Rahman, Shantanu Pal, Keerat Kaur, Robin Doss, and Chandan Karmakar. A privacy-preserving data integrity verification approach for access control in edge computing. In *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, 2025.

[14] Graeme Proudler, Liqun Chen, and Chris Dalton. *Trusted Computing Platforms: TPM2.0 in Context*. Springer, 2015.

[15] Vincent C. Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to attribute based access control (abac) definition and considerations. In *NIST Special Publication*, 2014.

[16] Xiang Gao, Jia Yu, Wen-Ting Shen, Yan Chang, Shi-Bin Zhang, Ming Yang, and Bin Wu. Achieving low-entropy secure cloud data auditing with file and authenticator deduplication. *Information Sciences*, 546:177–191, 2021.

[17] Yuefeng Du, Huayi Duan, Anxin Zhou, Cong Wang, Man Ho Au, and Qian Wang. Enabling secure and efficient decentralized storage auditing with blockchain. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3038–3054, 2022.

[18] Christos Tsigkanos, Marcello M. Bersani, Pantelis A. Frangoudis, and Schahram Dustdar. Edge-based runtime verification for the internet of things. In *2021 IEEE World Congress on Services (SERVICES)*, pages 16–16, 2021.