Anonymous Integrity Auditing Scheme based on Trusted Execution Environment for Distributed Edge Computing

Qingyang Zhang, Junjie Liu, Jie Cui, Fengqun Wang, Jiaxin Li, and Hong Zhong

***Abstract*—Multi-replica data storage is widely adopted in edge computing to improve both data availability and access efficiency for latency-sensitive applications. Integrity auditing is a critical mechanism for ensuring data reliability in this distributed setting. However, existing schemes face a trade-off between security and efficiency: ensuring replica authenticity typically requires users to generate unique tags for all copies, creating a bottleneck for resource-constrained devices. Delegation schemes reduce this burden but struggle to prevent collusion or on-the-fly generation attacks. Therefore, this study proposes ATRIA to resolve this dilemma. Uniquely, ATRIA leverages the Trusted Execution Environments (TEEs) not only for isolation but to securely offload the intensive replica tag generation from the user, ensuring authentic physical storage with minimal user overhead. By leveraging the hardware isolation of the TEEs, this mechanism ensures authentic physical storage and protects against on-the-fly and collusion attacks. In addition, the scheme incorporates a privacy-preserving identity management solution that balances anonymity and traceability. It employs a traceable anonymous identity mechanism whereby users interact via pseudonyms, hiding their real identities while allowing a trusted Key Generation Center (KGC) to perform identity tracing only when authorized. Our security analysis demonstrates that the proposed scheme achieves its security objectives and resists various attacks. Furthermore, a comprehensive performance evaluation demonstrates that ATRIA outperforms related schemes in terms of computational overhead.**

***Index Terms*—Edge Computing; Integrity Auditing; Trusted Execution Environment; User Anonymity; Multi-Replica Storage.**

I. INTRODUCTION

The rapid expansion of the Internet of Things (IoT) [1] and real-time interactive applications has exposed traditional cloud-computing models, particularly for their inherently high latencies that create bottlenecks. Thus, edge computing [2], [3] has become a potential countermeasure. It decentralizes computing and storage resources to the network edge in proximity to data sources or end users, thereby significantly reducing response latency through localized processing. Edge nodes are often required to cache and store multiple data replicates to ensure service continuity and high availability. However, once the data leaves the direct control of the user, security becomes a critical challenge. In open and complex edge environments, ensuring the completeness and integrity of distributed data replicas is a critical security requirement. Therefore, designing an efficient and secure data integrity auditing mechanism for distributed storage in edge computing (e.g., edge cache and edge storage) is a fundamental and core component for building a trustworthy edge ecosystem.

In multi-replica distributed storage scenarios [4]–[8] within edge environments, a trustworthy auditing mechanism [9] must satisfy several stringent security requirements. The first is the replica's authenticity and verifiability. The auditing process must achieve two critical objectives: verify data integrity (absence of tampering) and authenticate physical storage of independent replicas across edge nodes, preventing spoofing attacks where nodes dynamically generate proofs from a single copy [10]. Additionally, robust privacy preservation mechanisms [4] are essential to protect user data during these verification processes. During the public auditing process, the user's real identity must remain anonymous to unauthorized entities such as the auditor and edge nodes. Finally, the scheme must be user-friendly, meaning that the computational and communication overheads for the data owner (typically a resource-constrained end-user device) should be minimized, avoiding impracticality owing to burdensome cryptographic operations.

Existing research has explored various approaches [10]–[12] to address replica authenticity. The most direct method allows users to generate all replicas and their corresponding authentication tags. Although simple, this approach imposes a significant computational overhead on users, which is impractical for resource-constrained devices with numerous replicas. Another mainstream approach delegates replica generation to the storage nodes. However, this introduces the risk of an on-the-fly generation attack, in which a dishonest node only temporarily generates a replica upon receiving an audit challenge and does not occupy storage space at other times, which violates

Q. Zhang, J. Liu, J. Cui, F. Wang and H. Zhong are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China, and the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn). (Corresponding authors: Hong Zhong; Fengqun Wang.) J. Li is with Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei, Anhui, China and the Security Research Institute, New H3C Group, Hefei, 230088, China (email: li.jiaxin@h3c.com).

the principle of persistent data storage. To address this issue, Time-lock Puzzles [13], [14] have been proposed that require nodes to invest a specified amount of computation time to generate a valid replica. However, this, in turn, shifts the heavy computational burden to the edge nodes, which can severely affect their core service performance. Meanwhile, many traditional multi-replica Provable Data Possession (PDP) schemes [15]–[17] capable of auditing replicas often operate under a security model that assumes that replicas have been honestly generated, thus failing to effectively defend against collusion attacks.

Existing auditing schemes fail to sufficiently protect user identity privacy. The Traditional Public Key Infrastructure (PKI) [18] exposes user identities directly associated with public key certificates during an audit, failing to provide anonymity. Although Identity-Based Cryptography (IBC) [19], [20] simplifies certificate management, it suffers from the key-escrow problem [21] and typically uses the user's identity directly as a public key, thus failing to achieve privacy protection. Subsequent studies introduced excessive system overheads to achieve privacy, thereby limiting their practicality. Therefore, achieving traceable and lightweight anonymous identity protection while ensuring audit effectiveness remains a pressing problem.

This study proposed ATRIA, an anonymous auditing scheme for edge environments, to resolve the dual challenges of replica authenticity and user privacy. Our approach employs a Trusted Execution Environment (TEE) [22], [23] to enforce the authentic physical storage of replicas using unique hardware-generated obfuscation factors. This is combined with a traceable pseudonym mechanism that shields user identities from public view, while allowing authorized tracing by the KGC. The contributions of this study can be summarized as follows:

- **Efficient and Secure TEE-based Replica Assurance Mechanism**: To ensure the authentic physical storage of replicas, defend against on-the-fly and generation attacks, and simultaneously achieve user friendliness by alleviating the user's computational burden, this study proposes a novel TEE-based replica assurance mechanism. This mechanism leverages the TEE to generate replicas and their corresponding tags, effectively offloading intensive computations from users.
- **Privacy-Preserving Identity Management**: To address the challenge of protecting user identity during public audits, while maintaining accountability, we designed a privacy-preserving identity management scheme. This is achieved through a traceable anonymous identity mechanism that shields the user's real identity yet allows the KGC to perform identity tracing under authorized circumstances.
- **Formal Security Proof and Performance Evaluation**: We provide a rigorous security analysis to prove the security of the proposed scheme under the proposed security model, and through comprehensive performance evaluation experiments, we demonstrate its efficiency and practicality in terms of computational and communication overhead.

The structure of this paper is outlined below. We begin by surveying related work in Section II. Section III covers the essential background knowledge, while Section IV defines our system model and security framework. The concrete design of our ATRIA scheme is detailed in Section V, followed by its in-depth security proof in Section VI. In Section VII, we conduct a thorough performance analysis to show the scheme's practicality. The paper culminates in a conclusion in Section VIII.

II. RELATED WORK

Remote data integrity auditing is a core technology for ensuring the security of outsourced data. As application scenarios evolve towards the edge and multi-replica deployments, traditional cloud auditing schemes face new challenges. For edge scenarios, early research such as the ICE protocol proposed by Tong et al. [24], [25] primarily focused on user privacy protection. By combining Provable Data Possession (PDP) [26] and Proofs of Retrievability (PoR) [27] techniques, it allows a Third-Party Auditor (TPA) to complete audits without leaking user access patterns. Subsequent research shifted more towards the perspective of application providers, aiming to solve the challenges of audit efficiency and locating corrupted replicas across large-scale edge nodes. For instance, Li et al. and Cui et al. successively proposed schemes like EDI-V [28] and ICL-EDI [29]. These schemes introduced technologies such as the variable Merkle Hash Tree (VMHT) or homomorphic tags to enhance the efficiency of centralized auditing. However, to better adapt to the distributed nature of edge environments, subsequent research began to explore more lightweight and decentralized methods. Among them, the EDI-S scheme by Li et al. [30] utilizes aggregate signatures to significantly reduce verification overhead, while the CooperEDI scheme [31] implements a collaborative self-auditing and repair mechanism among edge nodes through a consensus mechanism. Recently, the research perspective expands to complex multi-vendor, multi-server scenarios. For example, the MVMS-SC framework proposed by Zhao et al. [32] prioritizes auditing high-risk replicas through smart contracts and intelligent detection algorithms, achieving more efficient and fair audits. These works collectively reflect the evolution of edge data auditing from simple integrity verification towards a more intelligent, efficient, and distributed comprehensive security framework.

In multi-replica storage scenarios [33]–[36], ensuring the authenticity and independence of replicas is the core of auditing. The foundational work in this area is the MR-PDP scheme proposed by Curtmola et al. [33], which uses a set of shared homomorphic verification tags for all replicas while distinguishing each replica with a unique random mask, thus enabling efficient batch verification. Subsequent research introduces numerous optimizations based on this foundation. To address the complexity of traditional Public Key Infrastructure (PKI), Li et

al. [35] proposed an identity-based multi-replica PDP scheme, emphasizing the distribution of replicas across multiple clouds to enhance security and efficiency. Guo et al. [37] optimized dynamic operations through the DPDPR scheme, using a shared authentication tree to reduce storage overhead. To resolve trust issues in multi-cloud environments, Zhang et al. [17] integrated blockchain technology [38]–[40] to record evidence and automate arbitration via smart contracts. Furthermore, researchers also pay close attention to user-friendliness. For example, the Mirror scheme by Armknecht et al. [10] transfers the replica construction task to the service provider, while Shen et al. [12] later proposed a proof-of-retrievability protocol with user-friendly replication, which balances the reduction of user burden and the guarantee of public verifiability through a novel replication algorithm with near-zero client-side computation and fixed communication costs.

In summary, existing schemes for both general multi-replica auditing and specific edge scenario auditing make significant progress in terms of efficiency, dynamics, and security. However, as stated in our introduction, most of them have not adequately addressed two core issues simultaneously: first, how to fundamentally prevent service providers from deceiving audits through collusion or on-the-fly generation of replicas, thereby ensuring the authentic physical storage of replicas, while also reducing the computational overhead on the user end to be user-friendly; and second, how to protect user identity privacy during the audit process. This is precisely the starting point and main contribution of our research.

III. PRELIMINARIES

A. Notations

We list the notations to be used in this paper in Table I.

TABLE I
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
G_1, G_T	Multiplicative cyclic groups
Z_q^*	$\{1, 2, \ldots, q-1\}$
g	A generator of G_1
q	The prime order of groups G_1, G_T
H_1, H_2, H_3, H_4	Hash functions
msk, mpk	Master secret/public key
RID	The real identity of Data Owner
(PID_1, PID_2)	The false identity of Data Owner
sk_u, pk_u	Secret key/public key of DO
$\{u_j = g^{a_j}\}_{j=1}^{s}$	Public values
F	The raw file
m	The copy number
$b_{i,j}$	The j-th sector of the i-th block of F
$b_{i,j,k}^*$	The k-th blinded replica of sector $b_{i,j}$
ϕ_u	Pseudorandom function (PRF) with key u
n	The block number
s	The sector number
σ_i	The signature of b_i
$\sigma_{i,k}$	The signature of The k-th copy of b_i
$chal = (i, v_i)$	The challenge from the TPA

B. Bilinear Pairing

A bilinear map is defined by a tuple (p, G_1, G_2, e), where:

- p is a large prime number.
- G_1 and G_2 are multiplicative cyclic groups of order p.
- $e : G_1 \times G_1 \to G_2$ is a bilinear pairing with the following properties:
  1) Bilinearity: For all $x, y \in G_1$ and $a, b \in \mathbb{Z}_p$,
  $$e(x^a, y^b) = e(x, y)^{ab}.$$
  2) Non-degeneracy:
  $$e(g, g) \neq 1_{G_2}, \quad \text{where } g \text{ is a generator of } G_1.$$
  3) Computability: For any randomly chosen elements $g_1, g_2 \in G_1$, the pairing $e(g_1, g_2)$ can be efficiently computed.

C. Hard Problems

- Computational Diffie-Hellman (CDH) Problem
  Let G be a multiplicative cyclic group of prime order p with generator g. Given a tuple $(g, g^a, g^b) \in G^3$ where $a, b \xleftarrow{R} \mathbb{Z}_p^*$, the CDH problem requires computing $g^{ab} \in G$. Formally, the CDH assumption holds in G if for every probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the following advantage is negligible:
  $$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}} = \Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab} \mid a, b \xleftarrow{R} \mathbb{Z}_p^*\right] \leq \mathsf{negl}(\lambda),$$
  where $\mathsf{negl}(\lambda)$ denotes a negligible function in the security parameter λ.

D. Trusted Execution Environment

A Trusted Execution Environment [41] is a hardware-enforced secure computing architecture that establishes isolated execution domains within general-purpose processors, independent of conventional operating systems. This is realized using hardware security extensions, with prominent examples being ARM TrustZone and Intel Software Guard Extensions (SGX) [42], [43], TEEs leverage physical isolation mechanisms at the silicon level to protect sensitive code and data. Memory protection units (MPUs) and address space controllers create hardware barriers between secure enclaves and untrusted environments—for instance, ARM TrustZone partitions mobile processors into secure and normal worlds to isolate critical operations like biometric authentication, while Intel SGX encrypts enclave memory and enforces strict access policies to prevent unauthorized access even from privileged system software. Persistent data protection is achieved through hardware-bound encryption keys derived from device-unique root secrets, coupled with security fusion techniques that cryptographically bind keys to specific hardware configurations, rendering physical extraction attacks infeasible. TEEs further ensure runtime integrity via secure boot chains and remote attestation protocols, which validate firmware authenticity and enclave contents before execution. By enforcing least-privilege access controls and

minimizing trusted computing bases, TEEs enable secure execution of cryptographic operations, digital rights management, and privacy-preserving machine learning in cloud and edge environments. This hardware-rooted model mitigates software vulnerabilities and supply chain risks while maintaining backward compatibility with legacy systems, positioning TEEs as a foundational technology for modern confidential computing paradigms.Specifically, ATRIA leverages Remote Attestation to ensure the integrity of the auditing code, and utilizes Hardware Isolation to shield sensitive keys (i.e., sk_{TEE} and u) from the untrusted host environment.

IV. System Model and Definitions

A. System Model

ATRIA comprises five major entities, namely: Key Generation Center (KGC), Data Owner (DO), Third-Party Auditor (TPA) , Cloud Storage Server (CSS) and Edge Storage Nodes (ESNs). The interaction among them is illustrated in Fig. 1.

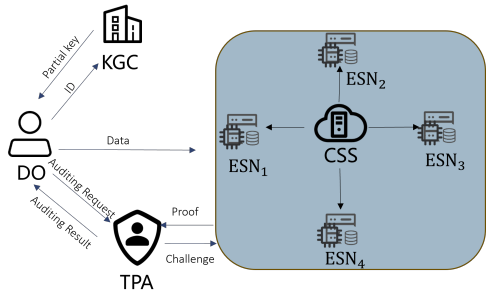


Fig. 1. The system model.

- **Key Generation Center (KGC):** The trusted authority responsible for generating system parameters and partial private keys for other entities in the system.
- **Data Owner (DO):** An entity possessing substantial volumes of data that require secure outsourcing to storage facilities.
- **Third-Party Auditor (TPA):** An independent party delegated by the DO to perform integrity verification of the outsourced data through regular audits.
- **Cloud Storage Server (CSS):** A centralized storage provider offering scalable storage capacity. The CSS is responsible for distributing user data to edge nodes and maintaining redundant copies for enhanced reliability.
- **Edge Storage Nodes (ESNs):** Distributed storage units deployed at network edge locations, which may include enterprise servers or base stations. These nodes facilitate low-latency data access for nearby terminal devices while providing localized storage services.It is worth noting that, to defend against collusion attacks, the replicas generated by the scheme must be mutually distinguishable, as identical replicas would create an opportunity for malicious nodes to share storage in order to pass the verification.

Trust Assumptions: We assume the KGC and DO are fully trusted, while the TPA is 'honest-but-curious'. The CSS and ESNs are treated as malicious entities capable of collusion or on-the-fly generation attacks. In contrast, the TEE embedded in the edge node serves as a trusted anchor protected by hardware isolation.

B. Scheme Definition

The proposed scheme ATRIA consists of the following eight algorithms.

Setup$(1^\kappa) \to (\text{msk}, \text{mpk}, \text{par})$: The KGC initializes the system by generating bilinear groups (G_1, G_T, e), a master secret key $\text{msk} = a$, a master public key $\text{mpk} = g^a$, cryptographic hash functions (H_1, H_2, H_3, H_4).

KeyGen$(\text{par}, \text{RID}) \to (\text{sk}_u, \text{pk}_u)$: A user generates pseudonyms $\text{PID}_1, \text{PID}_2$ for anonymous identity binding. The KGC verifies the user's real identity RID and issues partial private keys. The TEE generates hardware-enforced keys $(\text{sk}_u, \text{pk}_u)$.

TagGen$(\text{par}, F, \text{sk}_u) \to \{\sigma_i\}$: The Data Owner (DO) partitions file F into blocks $\{b_{i,k}\}$, computes aggregated sector values $S_i = \sum_{j=1}^{s} a_j b_{i,j}$, and generates authentication tags $\sigma_i = \text{sk}_{u1} \cdot (w_i \cdot g^{S_i})^{\text{sk}_{u2}}$, where $w_i = H_3(\text{FID} \parallel i \parallel \text{RID})$.

RepGen$(\text{par}, b_{i,j}, \text{sk}_u) \to \{b^*_{i,j,k}\}$: The TEE generates blinded replicas using PRF-based blinding factors $r_{i,j,k} = \phi_u(i \parallel j \parallel k \parallel \text{FID})$. Each replica is computed as $b^*_{i,j,k} = (b_{i,j} + r_{i,j,k})$.

RepTagGen$(\text{par}, b^*_{i,j,k}, \text{sk}_u, \text{sk}_{\text{TEE}}) \to \{\sigma_{ik}\}$: The TEE and ESN collaboratively compute replica tags $\sigma_{ik} = \sigma_i \cdot \text{pk}_u^{\sum_{j=1}^{s} a_j r_{i,j,k}} \cdot H_4(k)^{\text{sk}_{\text{TEE}}}$, ensuring integrity and privacy.

Challenge: $((1^\kappa, n) \to Q)$ The TPA randomly selects c block indices $I \subseteq [1, n]$ and generates coefficients $\{v_i\}_{i \in I} \xleftarrow{R} \mathbb{Z}_q^*$, constructing the challenge set $Q = \{(i, v_i)\}$.

ProofGen$(Q, \{\sigma_{ik}\}, \{b^*_{i,j,k}\}) \to (\sigma, \{M_k\}, \eta)$: ESNs aggregate signatures $\sigma = \prod_{k=1}^{m} \prod_{i \in I} \sigma_{ik}^{v_i}$, compute masked linear combinations $M_j = \sum_{k=1}^{m} \left(\sum_{i \in I} b_{jk} \cdot v_i + \lambda_{jk} \right)$, and generate bilinear verification parameters $\eta = \sum_{j=1}^{s} e(\prod_{k=1}^{m} g^{a_j \lambda_{jk}}, \text{pk}_u)$.

Verify$(\text{par}, \sigma, \{M_j\}, \eta) \to \{0, 1\}$: The TPA checks the equation,Verification succeeds if the equality holds.

C. Security Model

This section formally defines the security properties of the proposed ATRIA scheme. We begin by specifying the threat model, which details the potential actions of an adversary. Following this, we introduce the formal security definitions the scheme is designed to meet; these are captured via interactive games. All security proofs are based on the foundational assumption that the Computational Diffie-Hellman (CDH) problem is intractable in the group $\mathbb{G}_1$.

1) Threat Model: Our system considers two primary types of adversaries, which encapsulate the main threats to data integrity and user privacy in the edge environment. A clear definition of these adversaries is crucial for rigorously evaluating the robustness of our protocol.

- **Adversary $\mathcal{A}_1$ (External Attacker):** This adversary's goal is to forge a valid authentication tag for a data block without possessing the legitimate user's secrets. This represents an external entity attempting to compromise the integrity of the stored data, potentially to access data without authorization or to maliciously frame a legitimate user. We consider two powerful variations of this adversary:
  - $\mathcal{A}_{1a}$ can replace a user's public key with a value of its choice but does not know the system's master secret key (msk). This models an attacker who can manipulate public-facing parameters but has not compromised the central trust authority.
  - $\mathcal{A}_{1b}$ has compromised the KGC and therefore knows the msk, but it cannot modify users' public keys. This models a powerful insider threat.
- **Adversary $\mathcal{A}_2$ (Malicious ESN):** This adversary represents one or more Edge Storage Nodes that are malicious or have been compromised. Their primary goal is to deceive the TPA by generating a valid integrity proof for data that has been altered, deleted, or was never authentically stored. This model explicitly considers the possibility of collusion among multiple ESNs, reflecting realistic threats such as on-the-fly replica generation attacks.

2) Security Games and Definitions: We now define the security properties of the proposed scheme through three formal games. These games provide a standard methodology for proving security in modern cryptography.

a) Game 1: Tag Unforgeability: This game models the existential unforgeability of the proposed scheme against an adaptive chosen-message attack (EUF-CMA), capturing the threat posed by Adversary $\mathcal{A}_1$. The game proceeds between a challenger $\mathcal{C}$ and $\mathcal{A}_1$ as follows:

- **Setup**: The system is initialized by the challenger $\mathcal{C}$ through the **Setup** algorithm. Subsequently, the adversary $\mathcal{A}_1$ is furnished with the system parameters pp and the master public key mpk, which constitute the public environment. The master secret key msk, however, is withheld, save for the case against adversary $\mathcal{A}_{1b}$.
- **Queries**: To model a powerful adversary who can learn about the system, $\mathcal{A}_1$ is granted adaptive access to a set of oracles simulating the system's functionalities. These include oracles for hash computations, user key generation, revealing partial private keys, and generating tags for any chosen file under any user identity. For adversary $\mathcal{A}_{1a}$, an additional oracle for public key replacement is provided.
- **Forge**: Finally, $\mathcal{A}_1$ outputs a user identity PID^*, a file block b^*, and a forged tag σ^*. The adversary wins if σ^* constitutes a valid tag for block b^* under the identity PID^*, with the crucial condition that this specific tag was not previously obtained from the tag generation oracle during the query phase. A successful forgery means the adversary created a valid tag for a message it chose itself.

Definition 1 (Existential Unforgeability)**.** *The ATRIA scheme is considered existentially unforgeable if no probabilistic polynomial-time (PPT) adversary $\mathcal{A}_1$ can win Game 1 with more than a negligible probability.*

b) Game 2: Proof of Integrity: This game formalizes the integrity of the storage proof against malicious ESNs, as modeled by Adversary $\mathcal{A}_2$. The game demonstrates that an ESN cannot pass an audit without authentically storing the data.

- **Setup**: The challenger $\mathcal{C}$, acting as both the DO and TPA, generates all system parameters and keys. It then processes a file F to generate authentic replicas and their corresponding tags using the **RepGen** and **RepTagGen** algorithms, and distributes them to the adversary $\mathcal{A}_2$, who represents the untrusted storage.
- **Corruption**: To model real-world data loss or tampering, the adversary $\mathcal{A}_2$ is permitted to arbitrarily modify, corrupt, or delete any portion of the data replicas it holds.
- **Challenge**: $\mathcal{C}$ issues a random challenge Q to $\mathcal{A}_2$ for the stored file, requesting a proof of possession.
- **Forge**: $\mathcal{A}_2$ responds with a proof P. The adversary wins the game if the proof P successfully passes the verification check performed by $\mathcal{C}$, despite the underlying data being corrupted or incomplete. This would mean $\mathcal{A}_2$ has successfully broken the link between the proof and the actual state of the data.

Definition 2 (Proof of Integrity)**.** *The ATRIA scheme provides proof of integrity if for any PPT adversary $\mathcal{A}_2$, the probability of winning Game 2 is negligible.*

c) Game 3: User Anonymity: This game captures the property of user anonymity through a standard indistinguishability game, ensuring that user pseudonyms do not leak their real-world identities.

- **Setup**: The system parameters are generated by the challenger $\mathcal{C}$ using the **Setup** algorithm and are subsequently made public.
- **Challenge**: An adversary $\mathcal{A}$ chooses two distinct real identities, RID_0 and RID_1, and submits them to $\mathcal{C}$. The challenger randomly chooses a bit $b \in \{0,1\}$, computes $\mathsf{PID}^* \leftarrow \mathbf{KeyGen}(\mathsf{RID}_b)$, and then sends only the value PID^* to $\mathcal{A}$. The adversary's task is to determine if the pseudonym belongs to RID_0 or RID_1.
- **Guess**: The adversary $\mathcal{A}$ wins the game if it outputs a guess b' that correctly matches the challenger's chosen bit b, meaning $b' = b$.

Definition 3 (User Anonymity)**.** *The scheme provides user anonymity if for any PPT adversary $\mathcal{A}$, its advantage in winning Game 3, defined as $|\Pr[b' = b] - 1/2|$, is negligible. A negligible advantage implies that the adversary*

cannot distinguish which real identity corresponds to the given pseudonym with a success rate considerably higher than that of a random guess.

V. THE PROPOSED SCHEME

To address the dual challenges of integrity and privacy protection for multi-replica storage in edge computing environments, we propose an efficient, secure, and user-anonymity-supporting innovative auditing scheme named ATRIA. This scheme aims to resolve two core problems: replica authenticity and user privacy. We innovatively introduce a TEE to lead the replica generation process. Through the unforgeable obfuscation factors it produces, the proposed scheme guarantees the authentic physical storage of replicas at the hardware level, effectively defending against collusion and on-the-fly generation attacks. Simultaneously, the scheme designs a traceable anonymous identity mechanism, where users interact via pseudonyms to protect their privacy, and securely delegates heavy computational tasks to edge nodes and the TEE, significantly reducing the overhead on the user end and achieving user-friendliness. The following provides a detailed walkthrough of how the ATRIA scheme is constructed.

A. Setup

The Key Generation Center generates system parameters through the following steps:

1) Taking κ as the security parameter, the KGC chooses two cyclic multiplicative groups, G_1 and G_T, both of a prime order q, along with a generator $g \in G_1$ and a computable bilinear map $e\colon G_1 \times G_1 \to G_T$.
2) The KGC randomly selects the master secret key $msk = \alpha \xleftarrow{R} \mathbb{Z}_q^*$ and computes the master public key $mpk = g^\alpha \in G_1$.
3) The following cryptographic components are selected:
$$\begin{aligned} H_1&\colon G_1 \to \{0,1\}^l,\\ H_2&\colon \{0,1\}^* \to G_1,\\ H_3&\colon \{0,1\}^* \to G_1,\\ H_4&\colon \{0,1\}^* \to G_1, \end{aligned}$$
4) The KGC announces the system parameters $\mathsf{pp} = (q, g, \mathbb{G}_1, \mathbb{G}_T, e, \mathsf{mpk}, H_1, H_2, H_3, H_4)$.

B. KeyGen

The key generation protocol proceeds as follows:

1) A user with real identity $\mathsf{RID} \in \{0,1\}^l$ performs:
   - Randomly selects $z \xleftarrow{R} \mathbb{Z}_q^*$.
   - Constructs pseudonyms:
$$\begin{aligned} \mathsf{ID}_1 : \mathsf{PID}_1 &= g^z,\\ \mathsf{ID}_2 : \mathsf{PID}_2 &= \mathsf{RID} \oplus H_1(\mathsf{mpk}^z), \end{aligned}$$
   and sends $\mathsf{PID} = (\mathsf{PID}_1, \mathsf{PID}_2)$ to the KGC via a secure channel.
2) Upon receiving PID, the KGC:
   - Computes the partial private key:
$$\mathsf{sk}_{u1} = H_2(\mathsf{PID})^\alpha.$$
   - Can restore the user's real identity and trace it when necessary:
$$\mathsf{RID} = \mathsf{PID}_2 \oplus H_1(\mathsf{PID}_1^\alpha).$$
3) The user's complete private-public key pair is defined as:
   - Private key:
$$\mathsf{sk}_u = (\mathsf{sk}_{u1}, \mathsf{sk}_{u2}) = (H_2(\mathsf{PID})^\alpha, x),$$
   - Public key:
$$\mathsf{pk}_u = g^x.$$
4) TEE generates a corresponding key for each user for subsequent processing of the copy:
   - Private key: $\mathsf{sk}_{TEE} = t$, where $t \xleftarrow{R} \mathbb{Z}_q^*$.
   - Public key: $\mathsf{pk}_{TEE} = g^t$.
5) Key Establishment for PRF: We assume that the shared key u used for the PRF is securely provisioned by the DO to the TEE during the initial registration phase via a secure channel (established based on TEE Remote Attestation), ensuring that neither the OS nor other users can access it.

C. TagGen

In this phase, the user generates the authentication tags and transmits them to the storage provider. The process begins by partitioning the target file F into $n \times s$ data blocks, denoted by $b_{i,j}$ ($i \in [1,n]$, $j \in [1,s]$), where n stands for the total quantity of logical blocks and s represents the number of sectors per block, corresponding to the content of the j-th sector within the i-th block. Leveraging the user's private key $\mathsf{sk}_u = (\mathsf{sk}_{u_1}, \mathsf{sk}_{u_2})$ and a predefined set of secret coefficients:

$$\{a_j\}_{j=1}^s \xleftarrow{R} \mathbb{Z}_q^*,$$

These coefficients are generated by the DO and constitute part of the user's secret key material. They must be kept strictly confidential from the ESNs and the TPA to prevent the forgery of valid aggregated sector values.

The authentication tags are generated as follows: First, Combined with the secret coefficient to calculate the sector aggregation value S_i of the i-th block

$$S_i = \sum_{j=1}^{s} a_j \cdot b_{i,j}$$

Subsequently, generate the block-specific value w_i:

$$w_i = H_3\left(\mathsf{FID} \parallel i \parallel \mathsf{RID}\right),$$

where FID is the unique file identifier, RID represents the user's real identity. Finally, construct the authentication tag σ_i using the dual-private-key mechanism:

$$\sigma_i = \mathsf{sk}_{u_1} \cdot \left(w_i \cdot g^{S_i}\right)^{\mathsf{sk}_{u_2}}.$$

D. RepGen

In this phase, the TEE embedded within the edge node is responsible for generating blinding factors, which the edge storage node then uses to create the blinded versions of the data replicas. During the replica generation phase, the TEE ensures data block privacy protection and replica generation through the following steps:

1) The TEE generates blinding factors for each data block using a PRF based on the user's private key $u \in \mathbb{Z}_q^*$ (shared between the user and TEE):

$$r_{i,j,k} = \phi_u \left(i \parallel j \parallel k \parallel \mathsf{FID} \right),$$

ϕ_u: PRF parameterized by private key u, satisfying output indistinguishability from true randomness.

2) For the original data block $b_{i,j}$, the k-th blinded replica is generated as:

$$b^*_{i,j,k} = (b_{i,j} + r_{i,j,k})$$

This operation performs an arithmetic superposition of the original data $b_{i,j}$ and the blinding factor $r_{i,j,k}$ within the finite field $\mathbb{Z}_q$. In this manner, the generated replica $b^*_{i,j,k}$ exhibits randomness to an external observer, thereby effectively concealing the original data content. Despite the obfuscation of the data content, this algebraic structure retains crucial properties, enabling subsequent effective verification of its integrity without recovering the original data.

E. RepTagGen

In the Replica Tag Generation phase, the TEE and ESN collaboratively achieve secure generation and verification of multi-replica tags through hierarchical computation. The TEE first generates an intermediate value y based on the user's public key pk_u, a predefined secret coefficient set $\{a_j\}_{j=1}^s$, and its private key $sk_{\mathrm{TEE}} \in \mathbb{Z}_q^*$. Specifically:

$$y_k = pk_u^{\sum_{i=1}^s a_j \cdot r_{ijk}} \cdot H_4(k)^{sk_{\mathrm{TEE}}},$$

where the initialization factor r_{ijk} is produced by the pseudo-random function ϕ_u during the replica generation phase.

Upon receiving y, the ESN combines it with the authentication tag to compute the final replica tag:

$$\sigma_{ik} = \sigma_i \cdot pk_u^{\sum_{i=1}^s a_j \cdot r_{ijk}} \cdot H_4(k)^{sk_{\mathrm{TEE}}} = \sigma_i \cdot y_k.$$

Through algebraic transformation, it can be proven that:

$$\begin{aligned}
\sigma_{ik} &= sk_{u_1} \left(H_3(w_i) \cdot g^{\sum_{j=1}^s a_j \cdot b_{jk}} \right)^{sk_{u_2}} \cdot H_4(k)^{sk_{\mathrm{TEE}}} \\
&= sk_{u_1} \left(H_3(w_i) \cdot g^{\sum_{j=1}^s a_j (b_{ij} + r_{ijk})} \right)^{sk_{u_2}} \cdot H_4(k)^{sk_{\mathrm{TEE}}} \\
&= \sigma_i \cdot pk_u^{\sum_{j=1}^s a_j \cdot r_{ijk}} \cdot H_4(k)^{sk_{\mathrm{TEE}}} = \sigma_i \cdot y_k,
\end{aligned}$$

demonstrating that the replica tag preserves the integrity features of the original tag σ_i, while achieving privacy protection and multi-replica distinguishability through the initialization factor r_{ijk} and replica identifier $H_4(k)$. The term $H_4(k)^{sk_{TEE}}$ functions as a hardware-bound signature using the TEE's private key. By strictly binding the tag to the specific TEE instance and replica index k, it ensures replica uniqueness and mathematically precludes collusion attacks by malicious ESNs.

F. Challenge

During the integrity auditing phase, the TPA constructs a challenge set $Q = \{(i, v_i)\}$ through the following steps:

- A subset $I \subseteq [1, n]$, containing c distinct indices, is uniformly and randomly sampled from the index space $[1, n]$.
- For each selected index $i \in I$, a coefficient v_i is uniformly drawn from the finite field $\mathbb{Z}_q$, ensuring cryptographic unpredictability.
- The challenge set Q is formally defined as:

$$Q = \{(i, v_i) \mid i \in I\}, \quad \text{where } |I| = c.$$

The TPA subsequently transmits Q to the ESN to initiate the auditing procedure.

G. ProofGen

Upon receiving the challenge set from the auditor, the corresponding edge node executes the computation for this phase to generate a proof.The ESN generates an integrity proof based on the challenge set $Q = \{(i, v_i)\}$ provided by the TPA through the following steps:

1) For each storage server $k \in [1, m]$, the ESN computes an aggregated signature:

$$\sigma_k = \prod_{i \in I} \sigma_{ik}^{v_i},$$

where σ_{ik} denotes the original signature of data block b_{ik}, and $I \subseteq [1, n]$ is the challenge index subset. The global aggregated signature is then derived as:

$$\sigma = \prod_{k=1}^{m} \sigma_k.$$

This step leverages the multiplicative homomorphic property to enable efficient batch verification of multiple data block signatures.

2) To ensure data privacy and verifiability, the ESN constructs linear combinations using encoding coefficients $b^*_{i,j,k}$ and random masks $\lambda_{jk} \in \mathbb{Z}_q$. Specifically, the fragment parameter is generated as:

$$M_{jk} = \sum_{i \in I} b^*_{i,j,k} \cdot v_i + \lambda_{jk}.$$

Subsequently, these fragment parameters are aggregated into a global verification parameter:

$$M_j = \sum_{k=1}^{m} M_{jk} = \sum_{k=1}^{m} \left(\sum_{i \in I} b^*_{i,j,k} \cdot v_i + \lambda_{jk} \right).$$

The random mask λ_{jk} protects the privacy of the data by obfuscating it.

3) Utilizing the system public key pk_u and cryptographic generator g, the ESN computes bilinear pairing-based verification parameters. First, the fragment bilinear map value is derived as:

$$\eta_{jk} = e\left(g^{a_j \lambda_{jk}}, pk_u\right),$$

where a_j is a predefined secret coefficients. The fragment verification value is then obtained via product aggregation:

$$\eta_j = \prod_{k=1}^{m} \eta_{jk} = e\left(\prod_{k=1}^{m} g^{a_j \lambda_{jk}}, pk_u\right).$$

Finally, the global verification parameter is computed:

$$\eta = \prod_{j=1}^{s} \eta_j.$$

The ESN returns the triple $(\sigma, \{M_j\}_{j=1}^{s}, \eta)$ to the TPA.

H. Verify

In this phase, the TPA, upon receiving the storage proof, executes the verification process to confirm its validity.The TPA verifies the integrity proof by checking the following composite bilinear pairing equation:

$$\begin{aligned}\eta \cdot e(\sigma, g) = e\left(H_2(\mathrm{PID})^{\sum_{i\in I} v_i}, g^{\alpha}\right) \\ \cdot\ e\left(\prod_{k=1}^{m} H_4(k)^{\sum_{i\in I} v_i}, g^{\mathrm{sk_{TEE}}}\right) \\ \cdot\ e\left(\prod_{i\in I} H_3(\omega_i)^{v_i} \cdot \prod_{j=1}^{s} u_j^{M_j},\ pk_u\right)\end{aligned}$$

If the verification is successful, it proves that the storage node stores the data truthfully, and the program outputs 1, otherwise it outputs 0.

VI. SECURITY ANALYSIS

A. Correctness

This subsection demonstrates the correctness of our proposed scheme. Specifically, we will show that if the Data Owner (DO), Edge Storage Nodes (ESNs), and the Third-Party Auditor (TPA) all honestly execute the protocol, a proof generated from authentic and unmodified data will always pass the verification. The core of the correctness proof lies in the verification equation presented in the **Verify** algorithm. The proof proceeds by a step-by-step derivation from the left-hand side to the right-hand side. This derivation utilizes the foundational characteristics of bilinear pairings to establish the identity.

$$\begin{aligned}&\eta \cdot e(\sigma, g) \\ =&e\left(\prod_{j=1}^{s} u_j^{\sum_{k=1}^{m} \lambda_{jk}},\ pk_u\right) \cdot e\left(\prod_{k=1}^{m}\prod_{i\in I} \sigma_{ik}^{v_i},\ g\right) \\ =&e\left(\prod_{j=1}^{s} u_j^{\sum_{k=1}^{m} \lambda_{jk}},\ pk_u\right) \cdot e\left(\prod_{k=1}^{m}\prod_{i\in I} sk_{u1}^{v_i},\ g\right) \\ &\cdot e\left(\prod_{k=1}^{m}\prod_{i\in I}\left(H_3(w_i) \cdot g^{\sum_{j=1}^{s} a_j \cdot r_{ijk}}\right)^{sk_{u2}\cdot v_i},\ g\right) \\ &\cdot e\left(\prod_{k=1}^{m}\prod_{i\in I} H_4(k)^{sk_{\mathrm{TEE}}\cdot v_i},\ g\right) \\ =&e\left(\prod_{j=1}^{s} u_j^{\sum_{k=1}^{m} \lambda_{jk}},\ pk_u\right) \cdot e\left(H_2(PID)^{\sum_{i\in I} v_i},\ g^{\alpha}\right) \\ &\cdot e\left(\prod_{k=1}^{m} H_4(k)^{\sum_{i\in I} v_i},\ g^{sk_{\mathrm{TEE}}}\right) \\ &\cdot e\left(\prod_{k=1}^{m}\prod_{i\in I} H_3(w_i)^{v_i} \cdot \prod_{k=1}^{m}\prod_{j=1}^{s}\prod_{i\in I} u_j^{b_{ijk}\cdot v_i},\ pk_u\right) \\ =&e\left(H_2(\mathrm{PID})^{\sum_{i\in I} v_i}, g^{\alpha}\right) \cdot e\left(\prod_{k=1}^{m} H_4(k)^{\sum_{i\in I} v_i}, g^{sk_{\mathrm{TEE}}}\right) \\ &\cdot e\left(\prod_{i\in I} H_3(w_i)^{v_i} \cdot \prod_{j=1}^{s} u_j^{\sum_{k=1}^{m}(b_{ijk}\cdot v_i + \lambda_{jk})}, pk_u\right) \\ =&e\left(H_2(\mathrm{PID})^{\sum_{i\in I} v_i}, g^{\alpha}\right) \cdot e\left(\prod_{k=1}^{m} H_4(k)^{\sum_{i\in I} v_i}, g^{\mathrm{sk_{TEE}}}\right) \\ &\cdot e\left(\prod_{i\in I} H_3(\omega_i)^{v_i} \cdot \prod_{j=1}^{s} u_j^{M_j}, pk_u\right)\end{aligned}$$

B. Tag Unforgeability

Theorem 1 (Tag Unforgeability)**.** *Under the Computational Diffie–Hellman assumption in G_1, the advantage of any probabilistic polynomial-time adversary $\mathcal{A}_1$ in winning **Game 1** is at most negligible. Consequently, ATRIA satisfies tag unforgeability.*

We build a simulator $\mathcal{B}$ that receives a CDH instance (g, g^a, g^b) and outputs g^{ab}.

Setup. $\mathcal{B}$ sets the master public key $\mathsf{mpk} = g^a$ and embeds g^b into the hash oracle H_2 so that for the challenge identity $\mathsf{PID}^\star$ one has $H_2(\mathsf{PID}^\star) = g^b$. All other public parameters are generated honestly; $\mathcal{B}$ never needs the unknown master secret α.

Query phase. $\mathcal{A}_1$ may adaptively issue

- *H_i-queries ($i \in \{1,2,3,4\}$)* – answered at random;
- *KeyGen/Partial-Key/Secret-Value queries* – simulated with knowledge of α except for $\mathsf{PID}^\star$, whose partial key is *programmed* as $\mathsf{sk}^\star_{u1} = (H_2(\mathsf{PID}^\star))^\alpha = (g^b)^\alpha = g^{ab}$ *without* knowing g^{ab};
- *TagGen queries* – answered honestly, using the above keys and fresh randomness;
- (for $\mathcal{A}_{1a}$) *Public-Key-Replace* – $\mathcal{B}$ updates internal tables consistently.

Simulation is perfect from $\mathcal{A}_1$'s view since distributions match the real scheme.

Forgery. Eventually $\mathcal{A}_1$ outputs $(\mathsf{PID}^\star, b^\star, \sigma^\star)$ such that

- $(\mathsf{PID}^\star, b^\star)$ was *never* queried to the *TagGen oracle*;
- $\sigma^\star$ verifies under $\mathsf{PID}^\star$.

By definition of tag validity we have

$$e(\sigma^\star, g) = e\big(H_2(\mathsf{PID}^\star), g^\alpha\big)\, e\big(H_3(\omega^\star)\, g^{S^\star}, g\big)^{\mathsf{sk}^\star_{u2}}.$$

Substituting $H_2(\mathsf{PID}^\star) = g^b$ and cancelling known factors gives $g^{ab} = e(\sigma^\star/g^\delta, g)^{-1}$, where δ is fully computable by $\mathcal{B}$. Hence a successful forgery yields g^{ab} directly, so

$$\Pr\big[\mathcal{B} \text{ solves CDH}\big] \;\geq\; \Pr\big[\mathcal{A}_1 \text{ wins Game 1}\big].$$

Because CDH is hard, the latter probability is negligible.

C. Soundness of Integrity Proof

Theorem 2 (Proof of Integrity)**.** *Under the CDH assumption, the probability that any PPT adversary $\mathcal{A}_2$ wins **Game 2** is negligible. Hence ATRIA guarantees proof-of-integrity.*

Assume, for contradiction, that $\mathcal{A}_2$ outputs a forged proof $P = (\sigma, \{M_j\}, \eta)$ that passes verification for some corrupted replica set. We show how to build $\mathcal{B}$ that solves CDH.

Forking strategy. $\mathcal{B}$ embeds a CDH instance in H_4 and in the challenge indices v_i, and programs random oracles as in Theorem 1. Running $\mathcal{A}_2$ once, $\mathcal{B}$ records the transcript and rewinds it (Forking Lemma) with a different random challenge $Q' = \{(i, v_i')\}$. If both proofs P and P' are accepted, we obtain

$$e(\sigma/g^{\Delta}, g) = e(\sigma'/g^{\Delta'}, g),$$

which yields $g^{ab} = (\sigma/\sigma')^{1/(v_i - v_i')}$. Thus a non-negligible forging probability translates into a non-negligible CDH advantage, contradicting the assumption.

To clarify the distinction between the two theorems: Theorem 2 demonstrates that the ESN cannot pass verification with tampered or corrupted data, while the following Theorem 3 further proves that the ESN cannot forge a valid proof based on incomplete or (via collusion) not authentically stored replicas.

D. Unforgeability of Proof

Theorem 3 (Unforgeability of Proof)**.** *For the system setup, λ is the security parameter, while q denotes the prime order of the bilinear groups G_1, G_T. Assume that*

(i) the Discrete Logarithm (DL) problem in G_1 is hard, and

(ii) the secret coefficients $\{a_j\}_{j=1}^s \xleftarrow{\$} \mathbb{Z}_q^*$ used to define $u_j := g^{a_j}$ are chosen uniformly at random and kept hidden from the storage servers.

For every probabilistic polynomial–time (PPT) adversary $\mathcal{A}$ that does not hold all correct blinded replicas, its advantage

$$\mathbf{Adv}_{\mathcal{A}}^{\mathsf{forge}}(\lambda) \;=\; \Pr\big[\mathit{Verify}(P', \mathit{pk}, \mathit{params}) = 1\big]$$

in outputting a forged proof $P' = (\sigma', \{M_j'\}_{j=1}^s, \eta')$ is at most negligible in λ. Therefore the ATRIA scheme is unforgeable.

Proof. We use the standard sequence-of-games technique and focus on the final **Game 3**, in which the challenger issues a random challenge set $Q = \{(i, v_i)\}_{i \in I}$ and the adversary, controlling a set of cloud storage servers (CSSs), outputs a proof $P' = (\sigma', \{M_j'\}, \eta')$. The verifier accepts P' iff the following pairing equation holds:

$$\begin{aligned}\eta' \cdot e(\sigma', g) = {} & e\big(H_2(\mathsf{PID})^{\sum_{i\in I} v_i}, g^\alpha\big) \\ & \cdot e\Big(\textstyle\prod_{k=1}^m H_4(k)^{\sum_{i\in I} v_i}, g^t\Big) \\ & \cdot e\Big(\prod_{i\in I} H_3(\omega_i)^{v_i} \cdot \prod_{j=1}^s u_j^{M_j'},\; g^x\Big). \qquad (1)\end{aligned}$$

a) Deriving a binding condition.: Let $(\sigma, \{M_j\}, \eta)$ be the *honest* values that would be computed from the intact replicas. Define $\Delta M_j = M_j - M_j' \in \mathbb{Z}_q$ and $\Delta\mu = \sum_{j=1}^s a_j \Delta M_j$. Because the adversary has altered at least one replica, there exists a j^* with $\Delta M_{j^*} \neq 0$, so $\Delta\mu \neq 0$ with probability 1.

Plugging the honest values into (1) and cancelling equal terms, we obtain the necessary condition

$$\begin{aligned} e\big(g^{\Delta\mu}, g^x\big) = 1_{G_T} & \\ \iff \quad & g^{\Delta\mu} = 1_{G_1} \\ \iff \quad & \sum_{j=1}^s a_j \Delta M_j \equiv 0 \pmod q. \qquad (*)\end{aligned}$$

b) Probability analysis.: From the adversary's point of view, the tuple $(a_1, \ldots, a_s) \in (\mathbb{Z}_q^*)^s$ is distributed uniformly and independently of its view. The non-trivial linear relation $(*)$ therefore holds with probability at most $1/q \leq 2^{-\lambda}$, which is negligible.

c) Security reduction.: If $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{forge}}(\lambda)$ were non-negligible, one could distinguish whether a uniformly random s-tuple $(a_1, \ldots, a_s)$ satisfies a given non-trivial relation, contradicting assumption (ii) and ultimately allowing the solver to compute discrete logarithms in G_1 (violating assumption (i)).

Hence $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{forge}}(\lambda) \leq 1/q = \mathsf{negl}(\lambda)$, completing the proof. □

VII. PERFORMANCE EVALUATION

A. Functional Comparison

We perform a comparison between our method and a number of existing approaches, considering six critical features that are essential for a practical and secure edge data auditing solution. As shown in Table II, these features include: public audit, certificateless design, multi-copy support, replica consistency, user-friendliness, and identity privacy.

Each feature plays a vital role. **Public Audit** enables a data owner to offload the computationally intensive

TABLE II
FUNCTIONAL COMPARISON OF EXISTING MULTI-COPY SCHEMES

Scheme	Public Audit	Certificateless	Replica Consistency	User Friendly	Identity Privacy
MDSS [34]	Yes	Yes	Inconsistent	No	No
ICE [25]	Yes	No	Consistent	No	No
CLPDP-MCMS [36]	Yes	Yes	Inconsistent	No	No
ICL-EDI [29]	No	No	Consistent	No	No
The Proposed Scheme	Yes	Yes	Inconsistent	Yes	Yes

TABLE III
COMPUTATION OVERHEAD COMPARISON OF DIFFERENT SCHEMES

Scheme	MDSS [34]	CLPDP-MCMS [36]	The Proposed Scheme
TagGen	$2mn\text{Exp}_{G_1} + (2mn + m - 1)\text{Mul}_{G_1}$	$2mn\text{Exp}_{G_1} + (2mn + m - 1)\text{Mul}_{G_1}$	$2n\text{Exp}_{G_1} + 2n\text{Mul}_{G_1}$
RepTagGen	N/A	N/A	$mn\text{Exp}_{G_1} + 2mn\text{Mul}_{G_1}$
ProofGen	$c\text{Exp}_{G_1} + m(c-1)\text{Mul}_{G_1}$	$(c+m)\text{Exp}_{G_1} + mc\text{Mul}_{G_1}$	$mc\text{Exp}_{G_1} + ms\text{pair} + m(c+s)\text{Mul}_{G_1}$
Verify	$3\text{pair} + (mc + s + 1)\text{Exp}_{G_1} + (mc + s)\text{Mul}_{G_1}$	$3\text{pair} + (mc + s + 1)\text{Exp}_{G_1} + (mc + s - 1)\text{Mul}_{G_1}$	$4\text{pair} + (m + c + s + 3)\text{Exp}_{G_1} + (m + c + s + 1)\text{Mul}_{G_1}$

task of auditing to a Third-Party Auditor (TPA). A **Certificateless** architecture avoids the complex certificate management of the challenges posed by Public Key Infrastructure (PKI) and the built-in key escrow problem found in identity-based cryptography. **Multi-Copy** support is fundamental for ensuring data availability and fault tolerance. **Replica Consistency** is crucial for security, referring to the scheme's ability to guarantee that each replica is an authentic, independently stored copy, thereby fundamentally preventing on-the-fly generation and collusion attacks from malicious servers. Finally, a **User-Friendly** design minimizes the computational overhead for resource-constrained end-users, additionally, the function of **Identity Privacy** is to guarantee that the user's true identity is not revealed in public engagements.

As detailed in Table II, our proposed scheme is unique in its ability to satisfy the complete set of six features, offering a robust and well-rounded solution. In contrast, other schemes show significant limitations. For example, while typical multi-copy schemes like MDSS [34] and CLPDP-MCMS [36] support public auditing and multiple copies, they do not address the issues of user-friendliness or identity privacy. Other schemes designed for edge scenarios, such as ICE [25] and ICL-EDI [29], also fall short in providing a complete feature set, with notable gaps in multi-copy support, certificateless design, and particularly, in replica consistency. This comparative analysis highlights the significant advantages of the proposed scheme in providing a more secure, efficient, and a workable system for data integrity auditing in edge computing scenarios.

B. Performance Analysis

The computational and communication expenses of the protocol presented herein are analyzed in this section. In our assessment of computational overhead, the time required for one bilinear pairing operation is denoted by pair. The time taken to execute one multiplication operation within the group G_1 is represented by Mul_{G_1}, and the time consumed for an exponentiation operation within the group G_1 is signified by Exp_{G_1}. For the communication cost analysis, we define $|Z_q^*|$ and $|G_1|$as the respective bit-lengths of elements in the groups Z_q^* and G_1. Notably, ICE [25] and ICL-EDI [29] are omitted from the overhead comparison. They assume identical replicas, which, despite enabling highly efficient tag generation, creates a security vulnerability to collusion attacks among nodes. As the proposed scheme is fundamentally designed to prevent such attacks, a performance comparison would lack an equitable basis.

TABLE IV
COMMUNICATION COST COMPARISON

Scheme	Challenge	Proof
MDSS [34]	$c\|\mathbb{Z}_q^*\| + c * N$	$s\|\mathbb{Z}_q^*\| + \|G_1\|$
CLPDP-MCMS [36]	$c\|\mathbb{Z}_q^*\| + c * N$	$s\|\mathbb{Z}_q^*\| + \|G_1\|$
The Proposed Scheme	$c\|\mathbb{Z}_q^*\| + c * N$	$s\|\mathbb{Z}_q^*\| + \|G_1\| + \|G_T\|$

Computation Overhead: The main focus of this section is an assessment of the computational costs associated with the fundamental algorithms of our proposed scheme, specifically including the Tag Generation, Replica Tag Generation, Proof Generation, and Verification phases. For clarity, this analysis focuses on computationally intensive cryptographic operations, including exponentiation in group G_1 (denoted as Exp_{G_1}), multiplication in group G_1 (denoted as Mul_{G_1}), and bilinear pairing operations (denoted as pair). Table III provides a side-by-side com-

parison of the computational expenses for the aforementioned core algorithms between our proposed scheme and relevant comparative schemes. In this table, n represents the number of file blocks, m denotes the number of replicas, c is the number of challenged blocks, and s signifies the number of sectors per data block. Based on this table, the computational overhead characteristics of the proposed scheme are analyzed as follows: In the TagGen phase, the computational overhead of the proposed scheme is $2n\text{Exp}_{G_1} + 2n\text{Mul}_{G_1}$. When compared to the other schemes listed in the table, the overhead of our method is markedly lower. Since tag generation is only required for the original file, this feature is especially beneficial for resource-constrained users; tags related to replicas are subsequently generated with the assistance of the TEE, thereby reducing the computational capability requirements for the user, whereas other comparative schemes typically require the user to generate tags for all replicas. In the RepTagGen phase, the TEE is utilized to generate tags on a per-replica basis, with a computational overhead of $mn\text{Exp}_{G_1} + 2mn\text{Mul}_{G_1}$. This phase is specific to the proposed scheme, and other comparative schemes in the table usually do not involve this stage. In the ProofGen phase, the computational overhead of the proposed scheme is $mc\text{Exp}_{G_1} + ms\text{pair} + m(c+s)\text{Mul}_{G_1}$. Compared to the comparative schemes in the table, this overhead might be slightly higher, which represents a marginal increase compared to the other two approaches, but the proposed scheme is user-friendly and protects user privacy during the key generation phase. For the verification process, the computational cost is directly dependent on a trio of variables: the number of replicas m challenged blocks c, and sectors s. The overhead for the proposed scheme is $4\text{pair} + (m + c + s + 3)\text{Exp}_{G_1} + (m + c + s + 1)\text{Mul}_{G_1}$. Regarding exponentiation and multiplication operations, their coefficients are linearly related to $(m+c+s)$, whereas the corresponding coefficients in the comparative schemes include mc product terms. The total verification cost of our approach becomes considerably less than that of the benchmark schemes in the table, particularly as the replica count m grows large.

Communication Cost: In our integrity scheme, the communication overhead is primarily generated in two phases: the communication overhead produced during the challenge generation phase and the communication overhead produced during the proof generation phase. A comparative analysis of the communication costs for the two phases is detailed in Table IV, with N denoting $\lceil \log_2 n \rceil$ for clarity. For challenge generation, all three schemes exhibit an identical overhead of $c|\mathbb{Z}_q^*| + c * N$, which scales with the number of challenged blocks c. In the proof generation stage, the overhead incurred by our approach, $s|\mathbb{Z}_q^*| + |G_1| + |G_T|$, is comparable to that of the other two methods.

C. Simulation Experiment

For the evaluation of the efficiency of the proposed scheme, we established a corresponding experimental environment and conducted tests. The experimental platform was based on the Ubuntu 20.04.6 operating system, with hardware configurations including an Intel(R) Core(TM) i7-10710U CPU @ 1.10GHz and 16.00GB RAM, which support Intel SGX trusted execution environment. All algorithms in the scheme were implemented using the C++ language, and related cryptographic operations were performed using the MIRACL SDK. Specifically, for the selection of bilinear pairings, we chose Type-1 pairings from the MIRACL library that support AES-128 bit security strength.

Computation Overhead: The computational cost associated with the four primary stages of our presented approach is evaluated in this section: Tag Generation, Tag Generation with Replica Tag Generation, Proof Generation, and Verification. The experimental results are compared with relevant comparative schemes.

We commence by assessing the computational expense within the Tag Generation phase for various schemes under different numbers of file blocks, denoted as n. As illustrated in Figure 2, while the tag generation overhead for all schemes increases linearly with n, our proposed scheme is significantly more efficient. Specifically, when the number of file blocks is large, the proposed scheme shows a reduction of approximately 60% compared to the MDSS [34] and CLPDP-MCMS [36]. Our theoretical analysis is corroborated by this significant enhancement, which stems from the fact that tag generation is performed only on the original file (incurring a cost of $2n \cdot \text{Exp}_{G1} + 2n \cdot \text{Mul}_{G1}$), a process unaffected by the number of replicas m. Consequently, the proposed scheme is more amicable for users with limited computational resources.

Considering that the proposed scheme includes a TEE-assisted Replica Tag Generation phase with a theoretical overhead of $mn \cdot \text{Exp}_{G1} + 2mn \cdot \text{Mul}_{G1}$, we combine the overhead of the TagGen and RepTagGen phases in the proposed scheme for a fairer comparison of the total initial tag setup overhead against the TagGen overhead of other schemes (which typically already includes tag generation for all replicas). As shown in Figure 3, even when including the RepTagGen overhead, the proposed scheme still demonstrates a clear advantage in the total tag setup phase. While the computational cost of this scheme scales linearly with the quantity of file blocks n, it is substantially less than that of the benchmark schemes. At a high number of file blocks, our total setup cost shows a reduction of nearly 29.8% compared to the other schemes.

Following this, we investigate how the computational overhead of the Proof Generation phase is affected by variations in the number of challenged blocks c. Figure 4 illustrates that our approach incurs a marginally greater computational cost during the ProofGen phase when compared to both MDSS [34] and CLPDP-MCMS [36]. This outcome is in agreement with the preceding theoretical discussion, where the ProofGen overhead for the proposed scheme is $mc \cdot \text{Exp}_{G1} + ms \cdot \text{pair} + m(c + s) \cdot \text{Mul}_{G1}$, which includes ms pairing operations, typically computa-

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3670134

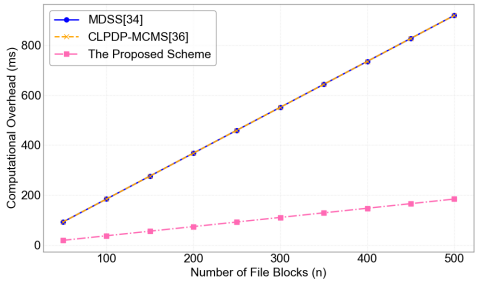


Fig. 2. Computational cost comparison of TagGen.

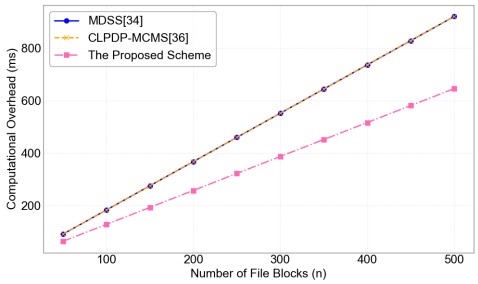


Fig. 3. Computational cost comparison of TagGen and RepTagGen.

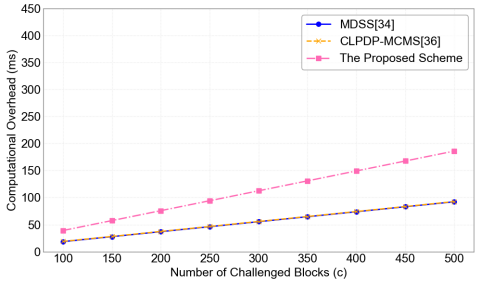


Fig. 4. Computational cost comparison of ProofGen.

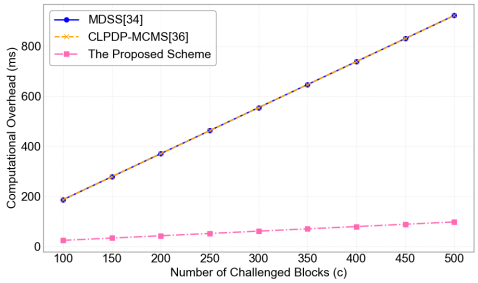


Fig. 5. Computational cost comparison of Verify.

tionally intensive. Admittedly, the overhead of our scheme is slightly elevated due to the pairing operations. However, this increased overhead represents a deliberate design trade-off. It is the reasonable cost paid for: completely offloading the tag generation overhead from the resource-constrained user side, and obtaining TEE-guaranteed replica authenticity to effectively resist collusion attacks. Thus, our framework successfully achieves the goal of being user-friendly for data owners.

Finally, we test the computational overhead of the Verification phase, while further investigating the effect of the challenged block count c. As shown in Figure 5, the proposed scheme demonstrates superior performance in the verification phase, thereby incurring a markedly reduced computational expense when compared to MDSS [34] and CLPDP-MCMS [36]. For a large number of challenged blocks, our verification overhead is reduced by approximately 59.3% compared to the other schemes. This advantage becomes more pronounced as c increases. According to our theoretical analysis, the overhead for the proposed scheme in this phase is $4\cdot\text{pair}+(m+c+s+3)\cdot\text{Exp}_{G1}+(m+c+s+1)\cdot\text{Mul}_{G1}$. The coefficients for exponentiation and multiplication operations are linearly related to $(m+c+s)$, whereas the corresponding coefficients in the comparative schemes include mc product terms. Despite the growth in the quantity of challenged blocks c, the experimental findings confirm that the verification expense of our method stays low.

Communication Cost: we conducts an experimental assessment of the communication cost of the presented approach in the challenge generation and proof generation stages. Within the challenge generation process, the presented approach, as well as MDSS [34] and CLPDP-MCMS [36], has a communication overhead amounting to $c|Z_q^*| + c$,. This value is predominantly dependent on the security parameter c.

In the proof generation phase, the proof size for the MDSS [34] and the CLPDP-MCMS [36] is $s|Z_q^*| + |G_1|$, while for the proposed scheme, it is $s|Z_q^*| + |G_1| + |G_T|$. We fix $s = 10$ and observe the communication overhead of each scheme as the number of replicas m varies from 1 to 10. Figure 6 illustrates that while the communication overhead of our scheme is slightly elevated compared to MDSS [34] and CLPDP-MCMS [36] at $s = 10$, the overhead for all schemes in the proof phase is unaffected by the number of replicas m when the sector count s is fixed.

In summary, our experimental results are highly consistent with the theoretical analysis. In terms of communication overhead, the proposed scheme performs similarly to the other schemes during the challenge phase. In the proof phase, despite the increased overhead in our scheme, which is caused by incorporating an additional G_T element, it remains constant with respect to the number of replicas. This is an advantageous characteristic for storage systems with a large number of replicas, as it avoids a linear increase in communication cost with the replica count.

VIII. CONCLUSION

Leveraging the TEE, this study puts forward a scheme for auditing edge data integrity, named ATRIA, which addresses two issues: integrity verification and user anonymity protection for multi-replica data storage in edge-computing environments. This scheme uses the hardware security features of the TEE to generate obfuscation factors for data replicas, ensuring the authentic storage

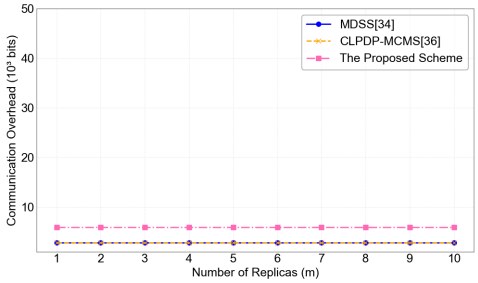


Fig. 6. Communication cost comparison of Proof.

and unforgeability of data replicas, and fundamentally enhancing the security of data storage. Furthermore, we designed and integrated an anonymous identity management mechanism into the key generation process, effectively protecting users' identity privacy during the auditing process while retaining the ability for identity tracing by the KGC under authorized circumstances. This scheme significantly reduces the computational overhead for users during tag generation and replica management, thereby improving user friendliness. Ultimately, the theoretical security and reliability of our proposed ATRIA scheme are confirmed by both a comprehensive security analysis and the results from detailed performance evaluations, and demonstrates high efficiency and feasibility in practical applications. For future work, we will focus on extending the ATRIA framework to support integrity auditing for dynamic data. Key challenges include ensuring distributed replica consistency under concurrent updates and achieving efficient tag updates without full reconstruction in TEEs.

References

[1] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of things (iot): A literature review," *Journal of computer and communications*, vol. 3, no. 5, pp. 164–173, 2015.

[2] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE access*, vol. 8, pp. 85 714–85 728, 2020.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[4] X. Zhang, X. Wang, D. Gu, J. Xue, and W. Tang, "Conditional anonymous certificateless public auditing scheme supporting data dynamics for cloud storage systems," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5333–5347, 2022.

[5] H. Duan, Y. Du, L. Zheng, C. Wang, M. H. Au, and Q. Wang, "Towards practical auditing of dynamic data in decentralized storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 708–723, 2022.

[6] J. Zhao, H. Huang, D. He, X. Zhang, Y. Zhang, and K.-K. R. Choo, "Ib-iadr: Enabling identity-based integrity auditing and data recovery with fault localization for multi-cloud storage," *IEEE Internet of Things Journal*, 2024.

[7] Q. Zhang, Z. Zhang, J. Cui, H. Zhong, Y. Li, C. Gu, and D. He, "Efficient blockchain-based data integrity auditing for multi-copy in decentralized storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 12, pp. 3162–3173, 2023.

[8] Q. Zhang, D. Sui, J. Cui, C. Gu, and H. Zhong, "Efficient integrity auditing mechanism with secure deduplication for blockchain storage," *IEEE Transactions on Computers*, vol. 72, no. 8, pp. 2365–2376, 2023.

[9] Y. Li and F. Zhang, "Remote data auditing for cloud-assisted wbans with pay-as-you-go business model," *Chinese Journal of Electronics*, vol. 32, no. 2, pp. 248–261, 2023.

[10] F. Armknecht, L. Barman, J.-M. Bohli, and G. O. Karame, "Mirror: Enabling proofs of data replication and retrievability in the cloud," in *25th USENIX security symposium (USENIX security 16)*, 2016, pp. 1051–1068.

[11] W. Guo, S. Qin, J. Lu, F. Gao, Z. Jin, and Q. Wen, "Improved proofs of retrievability and replication for data availability in cloud storage," *The Computer Journal*, vol. 63, no. 8, pp. 1216–1230, 2020.

[12] J. Shen, X. Chen, X. Huang, and Y. Xiang, "Public proofs of data replication and retrievability with user-friendly replication," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 2057–2067, 2023.

[13] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," 1996.

[14] G. Malavolta and S. A. K. Thyagarajan, "Homomorphic time-lock puzzles and applications," in *Annual International Cryptology Conference*. Springer, 2019, pp. 620–649.

[15] A. F. Barsoum and M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 485–497, 2014.

[16] J. Shen, P. Zeng, and K.-K. R. Choo, "Multicopy and multiserver provable data possession for cloud-based iot," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12 300–12 310, 2021.

[17] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A blockchain-based multi-cloud storage data auditing scheme to locate faults," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2252–2263, 2021.

[18] J. A. Buchmann, E. Karatsiolis, and A. Wiesmaier, *Introduction to public key infrastructures*. Springer Science & Business Media, 2013.

[19] M. Joye and G. Neven, *Identity-based cryptography*. IOS press, 2009, vol. 2.

[20] D. Anand, V. Khemchandani, and R. K. Sharma, "Identity-based cryptography techniques and applications (a review)," in *2013 5th international conference and computational intelligence and communication networks*. IEEE, 2013, pp. 343–348.

[21] T. H. Yuen, W. Susilo, and Y. Mu, "How to construct identity-based signatures without the key escrow problem," *International Journal of Information Security*, vol. 9, pp. 297–311, 2010.

[22] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *2015 IEEE Trustcom/BigDataSE/Ispa*, vol. 1. IEEE, 2015, pp. 57–64.

[23] A. Muñoz, R. Ríos, R. Román, and J. López, "A survey on the (in) security of trusted execution environments," *Computers & Security*, vol. 129, p. 103180, 2023.

[24] W. Tong, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-preserving data integrity verification in mobile edge computing," in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*. IEEE, 2019, pp. 1007–1018.

[25] W. Tong, W. Chen, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-preserving data integrity verification for secure mobile edge storage," *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5463–5478, 2022.

[26] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 598–609.

[27] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 584–597.

[28] B. Li, Q. He, F. Chen, H. Dai, H. Jin, Y. Xiang, and Y. Yang, "Cooperative assurance of cache data integrity for mobile edge computing," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4648–4662, 2021.

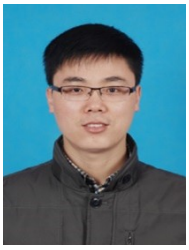
[29] G. Cui, Q. He, B. Li, X. Xia, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Efficient verification of edge data integrity in edge computing environment," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3233–3244, 2021.

[30] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Inspecting edge data integrity with aggregate signature in distributed

edge computing environment," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2691–2703, 2021.
[31] J. Li, Q. Zhao, H. Cheng, S. Teng, N. Wu, and Y. Liang, "Oredi: A per-edge one-round data integrity verification scheme for mobile edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 2, pp. 2074–2086, 2023.
[32] Y. Zhao, Y. Qu, Y. Xiang, F. Chen, M. P. Uddin, and L. Gao, "Winning at the starting line: Unreliable data replica selection for edge data integrity verification," *IEEE Transactions on Services Computing*, 2024.
[33] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "Mr-pdp: Multiple-replica provable data possession," in *2008 the 28th international conference on distributed computing systems*. IEEE, 2008, pp. 411–420.
[34] L. Zhou, A. Fu, G. Yang, H. Wang, and Y. Zhang, "Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1118–1132, 2020.
[35] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Transactions on Cloud Computing*, vol. 10, no. 1, pp. 356–365, 2019.
[36] J. Shen, P. Zeng, K.-K. R. Choo, and C. Li, "A certificateless provable data possession scheme for cloud-based ehrs," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1156–1168, 2023.
[37] W. Guo, S. Qin, F. Gao, H. Zhang, W. Li, Z. Jin, and Q. Wen, "Dynamic proof of data possession and replication with tree sharing and batch verification in the cloud," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 1813–1824, 2020.
[38] W. Liang, S. Xie, K.-C. Li, X. Li, X. Kui, and A. Y. Zomaya, "Mc-dsc: A dynamic secure resource configuration scheme based on medical consortium blockchain," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3525–3538, 2024.
[39] S. Shamshad, K. Mahmood, S. Kumari, C.-M. Chen *et al.*, "A secure blockchain-based e-health records storage and sharing scheme," *Journal of Information Security and Applications*, vol. 55, p. 102590, 2020.
[40] F. Wang, J. Cui, Q. Zhang, D. He, C. Gu, and H. Zhong, "Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial internet of things," *IEEE transactions on dependable and secure computing*, vol. 21, no. 4, pp. 1587–1604, 2023.
[41] T. Geppert, S. Deml, D. Sturzenegger, and N. Ebert, "Trusted execution environments: Applications and organizational challenges," *Frontiers in Computer Science*, vol. 4, p. 930741, 2022.
[42] F. McKeen, I. Alexandrovich, I. Anati, D. Caspi, S. Johnson, R. Leslie-Hurd, and C. Rozas, "Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave," in *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*, 2016, pp. 1–9.
[43] S. Johnson, V. Scarlata, C. Rozas, E. Brickell, F. Mckeen *et al.*, "Intel software guard extensions: Epid provisioning and attestation services," *White Paper*, vol. 1, no. 1-10, p. 119, 2016.

Qingyang Zhang was born in Anhui Province, China, in 1992. He received his B. Eng. degree and Ph.D. degree in computer science from Anhui University in 2021. He is currently an associate professor of School of Computer Science and Technology at Anhui University. His research interest includes edge computing, computer systems, and security. He has over 30 scientific publications in reputable journals (e.g. Proceedings of the IEEE, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers) and international conferences.

Junjie Liu is now a research student in the School of Computer Science and Technology, Anhui University. His research focuses on the security of the cloud computing and edge computing.



Jie Cui (Senior Member, IEEE) was born in Henan Province, China, in 1980. He received his Ph.D. degree in University of Science and Technology of China in 2012. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). He has over 150 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers), academic books and international conferences.

Fengqun Wang was born in Anhui Province, China, in 1996. He received his Ph.D. degree in computer science from Anhui University in 2024. He is currently a lecture of School of Computer Science and Technology at Anhui University. His research interests include IoT security, blockchain and applied cryptography. He has multiple scientific publications in reputable journals (e.g. IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Network and Service Management, IEEE Transactions on Industrial Electronics).

Jiaxin Li currently holds the position of Director of Government Affairs at H3C Information Security Technology Co., Ltd. He obtained his master's degree from Anhui University and is pursuing a doctoral degree at the same institution. Li Jiaxin's research focuses on the security of vehicular ad hoc network, data security, and the security of Industrial Internet of Things.

Hong Zhong was born in Anhui Province, China, in 1965. She received her PhD degree in computer science from University of Science and Technology of China in 2005. She is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. Her research interests include applied cryptography, IoT security, vehicular ad hoc networks, cloud computing security and software-defined networking (SDN). She has over 200 scientific publications in reputable journals (e.g. IEEE Journal on Selected Areas in Communications, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security), academic books and international conferences.