This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

A Resilient Edge Data Integrity Verification Approach via Robust Merkle Trees and Steganographic Transmission

Md Rashedul Islam, *Member, IEEE*, Yong Xiang, *Senior Member, IEEE*, Md Palash Uddin, *Member, IEEE*, Yushu Zhang, *Senior Member, IEEE*, Dezhong Peng, *Senior Member, IEEE*, Jonathan Kua, *Member, IEEE*, and Abdorasoul Ghasemi

Abstract—Edge Data Integrity Verification (EDIV) plays a crucial role in maintaining the authenticity of cached data replicas in decentralized and resource-constrained edge computing environments. However, existing EDIV methods prioritize generating cryptographic proofs while overlooking the need to ensure robustness against localized data tampering and neglecting the inherent security risks involved in transmitting these proofs over untrusted backhaul communication networks. These gaps leave the EDIV process vulnerable to both the manipulation of integrity proofs and their interception during transmission, ultimately undermining the reliability of EDIV. To address these gaps, we propose RAMTStego-EDIV, a novel framework that ensures both the robustness of integrity proofs and their secure transmission. In particular, RAMTStego-EDIV first introduces the Robust Aggregated Merkle Tree (RAMT)—a ternary hash tree that combines real and auxiliary nodes to produce tamper-evident cryptographic proofs, ensuring that even partial manipulation of the data can be reliably detected. Then, it develops a dual-parity transmission control protocol (TCP) steganography technique to protect the transmission of the RAMT-driven cryptographic proofs that covertly embed the proof bits into two TCP header fields: the parity of the payload length and the least significant bit of the TCP window size. This embedding allows proofs to travel invisibly within normal traffic, avoiding detection and interference by adversaries monitoring the network. Both theoretical analysis and experimental results demonstrate that RAMTStego-EDIV prevents proof tampering and interception, confirming its resilience and suitability for deployment in adversarial mobile edge computing environments.

Index Terms—Edge Data Integrity, Edge Computing, Steganography, Aggregated Merkle Hash Tree.

1 Introduction

The explosive growth of edge devices, including smartphones, Internet of Things (IoT) sensors, and autonomous vehicles, has rapidly outpaced the capabilities of traditional computing infrastructures. Projections estimate that the number of such devices will surpass 20 billion by 2025 [1], creating significant challenges for efficient and reliable large-scale data processing. While cloud computing has served as a scalable backbone for handling increasing demands over the past decade, its centralized architecture is increasingly strained under the pressure of supporting latency-sensitive applications [2]. As a result, cloud-based systems often fail to meet the stringent performance and communication requirements posed by modern edge environments. Mobile Edge Computing (MEC) brings computation and storage closer to users, reducing latency and enabling real-time applications like augmented reality, autonomous driving, and industrial automation [3]. While MEC improves responsiveness by decentralizing processing, it also raises new concerns about the integrity of data distributed across diverse Edge Servers (ESs). Unlike centralized cloud systems, MEC involves heterogeneous ESs operated by multiple application vendors (AVs), making it challenging to ensure the trustworthiness of independently cached data replicas, a core issue in Edge Data Integrity (EDI) [4]–[6].

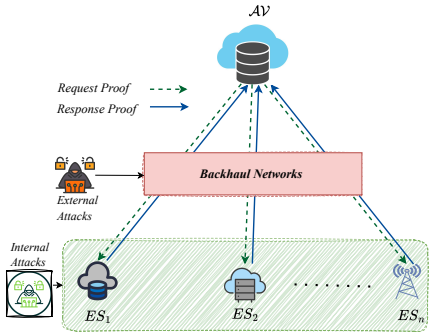


Fig. 1: An illustration of internal and external attacks in MEC.

Md Rashedul Islam, Yong Xiang, Md Palash Uddin, and Jonathan Kua are with the School of Information Technology, Deakin University, Geelong, VIC 3220, Australia. Email: {md.islam, yong.xiang, m.uddin, jonathan.kua}@deakin.edu.au
Y. Zhang is with the School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics, China, email: zhangyushu@jxufe.edu.cn
D. Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with Sichuan National Innovation New Vision UHD Video Technology Company Ltd., Chengdu 610095, China, email: pengdz@scu.edu.cn
A. Ghasemi is with the Centre for Computational Science and Mathematical Modelling, Coventry University, UK, email: ae4959@coventry.ac.uk
Corresponding Author: Yong Xiang
This work was supported in part by the Australian Research Council under grant LP190100594.

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

Existing EDI Verification (EDIV) approaches (see related work section 6) follow the classical challenge-response strategy and rely on key-based cryptographic techniques or lightweight sampling-based mechanisms applied over hashed data structures to validate the integrity of cached data on ESs [7]–[11]. Although these methods can detect large-scale data corruption, they often struggle to catch smaller or more targeted tampering, especially when the verification process does not tightly link all parts of the data together [12]–[14]. Furthermore, they commonly implicitly assume that the communication channel over which these proofs are exchanged, specifically the challenge-response messages, is secure [4], [7], [15], [16]. This assumption does not hold in real-world MEC environments, where the backhaul network between AVs and ESs frequently traverses untrusted or semi-trusted infrastructure such as public Internet links, 5G core networks, or shared wireless access points [17]. These channels are susceptible to interception and attacks [18]. In practice, MEC environments present two main attack surfaces that hinder the feasibility of EDIV: (i) the ES side, which may suffer from internal attacks, and (ii) the transmission channel, where *challenge-response* exchanges occur and are exposed to external attacks. These scenarios are illustrated in Fig. 1, from which it can be seen that external adversaries actively disrupt or manipulate communication between the AV and ES, targeting the delivery of integrity proofs. An adversary compromising this channel may alter challenge-response messages to generate false verification results, even if the underlying data is intact. This overlooked vulnerability reveals a critical blind spot in existing EDIV approaches. Without protection in the transmission channel, even strong cryptographic proof generation becomes ineffective. Therefore, a robust EDIV solution must secure not only the stored data and generated proofs but also the communication process that underpins the verification protocol. These observations raise two fundamental research questions that remain underexplored in current EDIV literature:

RQ1: *How can proof-generation mechanisms be strengthened so that even localized or selective tampering in cached data propagates more strongly through the verification structure, without increasing proof size or verifier cost?* ***RQ2:*** *How can the transmission of proofs over untrusted backhaul networks be secured in a manner that is both stealthy (hard to detect or block) and reliable (resistant to packet loss or manipulation), without incurring significant bandwidth or computation overhead?*

To address these research questions, we propose RAMTStego-EDIV, a novel framework for robust and secure EDIV that integrates a Merkle Hash Tree (MHT)-based structure with a steganographic technique to protect the verification process. Specifically, we propose an enhanced MHT structure called Robust Aggregated Merkle Tree (RAMT) for integrity-proof generation and a dual-parity transmission control protocol (TCP) steganographic method to secure proof transmission from ESs to AV. Unlike prior Merkle-tree enhancements that mainly target proof-size reduction, hashing efficiency, or construction optimization, RAMT is explicitly designed to improve tamper-propagation robustness under localized corruption. The RAMT significantly strengthens the robustness of proof generation by employing a unique ternary hashing mechanism. Unlike conventional binary or ternary Merkle Trees, where a leaf modification affects only a single ancestor path, RAMT incorporates an auxiliary cross-sibling hashing node at every level. This design creates additional dependency paths, effectively amplifying the visibility of tampering across a larger portion of the tree. Under practical partial-audit conditions, where verifiers examine only a small number of internal nodes per round, this amplified propagation significantly increases the likelihood of detecting targeted or low-frequency corruption. As a result, RAMT provides substantially improved detection efficiency and tamper exposure, addressing a key limitation of traditional Merkle-based integrity proofs in distributed edge environments.

Complementing the structural enhancement offered by RAMT, our dual-parity TCP steganography secures the often-overlooked transmission of integrity proofs. RAMTStego-EDIV embeds proof segments covertly into live TCP flows using two header fields: (i) the sequence-number parity (SNP), adjusted through payload-length increments, and (ii) the window-size parity (WSP), modified via the least significant bit of the TCP window field. Distributing embedding across both fields increases capacity and improves resistance to packet normalization and detection by network adversaries. The method introduces minimal overhead—typically limited to ±1-byte adjustments, and remains statistically indistinguishable from normal traffic, enabling practical deployment in resource-constrained and adversarial edge environments. By combining RAMT's tamper-resilient proof structure with a stealthy and reliable covert channel, RAMTStego-EDIV mitigates vulnerabilities in transmitting integrity proofs over untrusted networks. Our experiments confirm that this integration preserves proof integrity while maintaining low communication overhead, significantly strengthening EDIV in MEC settings.

Unlike prior work that treats data verification and transmission security as separate problems, this study provides a unified formulation of EDIV under adversarial backhaul conditions. RAMTStego-EDIV jointly integrates a tamper-resilient authenticated structure (RAMT) with a provably robust and stealthy transmission channel under a common adversarial model capturing both internal and external threats. Supported by formal theorems on robustness, unforgeability, reliability, and indistinguishability, the framework elevates EDIV from a heuristic practice to a theoretically grounded end-to-end integrity solution. In this formulation, the verification structure and its communication channel are protected simultaneously, ensuring they function as inseparable components of the same security objective. To our knowledge, no prior EDIV approach achieves this level of unification. Compared to existing MHT variants, RAMT amplifies tamper propagation without increasing proof size, while the dual-parity covert channel provides provable transmission security with negligible computation and bandwidth overhead. The main contributions of this work can be summarized as follows:

- **Robust Integrity Proofs via RAMT:** We introduce RAMT, an enhanced Merkle-based structure that increases tamper visibility and improves resilience against localized corruption in edge environments. Our analysis and experiments

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

show that RAMT provides stronger detection robustness than traditional MHTs while preserving logarithmic proof size and computational efficiency.
- **Secure and Covert Proof Transmission via Dual-Parity Steganography:** We identify a critical vulnerability in existing EDIV frameworks—proof packets remain exposed during transmission over untrusted networks. To address this gap, we propose a dual-parity TCP steganographic method that embeds proof bits across both sequence-number parity and window-size parity, offering greater stealth and robustness than prior single-field techniques. This significantly strengthens the end-to-end integrity assurance process in decentralized edge systems.

The remainder of the paper is organized as follows. Section 2 presents the system and threat models. Section 3 outlines the RAMTStego-EDIV framework and its workflow. Sections 4 and 5 provide the theoretical analysis and experimental evaluation. Section 6 reviews related work, and Section 7 concludes the paper.

2 System Model

We consider the common assumption that communication within the environment occurs through untrusted channels without additional delays [6] to present the system architecture, provide the threat model, and outline the design goals.

2.1 System Architecture

The MEC system consists of two main entities: an AV and a set of n edge servers $\mathcal{MES} = \{ES_1, ES_2, \ldots, ES_n\}$, where $n \geq 2$. The AV owns a dataset $D_i = \{d_1, d_2, \ldots, d_k\}$ consisting of k data items. The AV is assumed to be honest and trustworthy, meaning that it faithfully and periodically sends verification requests to each ES, verifies the returned proofs, and records the results.

Each ES_j stores a subset of the AV's dataset, denoted as $D_{i,j} \subseteq D_i$, and upon receiving a verification request must return a proof p_j attesting to the integrity of its cached data. We assume that the ESs operate independently without collusion, consistent with prior works [15], [19]. The problem is to design a verification framework that satisfies two conditions:
(i) For any modified data $D'_{i,j} \neq D_{i,j}$, even if only one block is altered, the AV should reject the corresponding proof: $\mathcal{V}(D'_{i,j}, p_j) = \text{Reject}$. (ii) When proof p_j is transmitted over an untrusted channel, any adversarial modification $p'_j \neq p_j$ must also be detected and rejected: $\mathcal{V}(D_{i,j}, p'_j) = \text{Reject}$.

2.2 Threat Model

The proposed RAMTStego-EDIV framework considers two primary adversarial threats within the MEC environment: (1) unreliable ESs, known as internal adversaries, and (2) untrusted communication channels, referred to as external adversaries. Table 1 summarizes the two adversarial classes considered in RAMTStego-EDIV, together with their locations, capabilities, and representative attack vectors. We assume that ESs operate independently without collusion, the AV remains trusted, and adversaries do not break the collision resistance of the underlying hash function.

TABLE 1: Summary of adversarial classes in the threat model.

Adversary Type	Location	Capabilities	Attack Vectors
Internal adversary (unreliable ES)	Edge server	Modify cached data; generate forged proofs; reuse stale proofs	Replay attacks; replacement attacks; forgery attacks
External adversary (network attacker)	Backhaul channel	Observe, intercept, alter, replay, inject, or drop proof-carrying packets	Proof tampering; man-in-the-middle interference; packet injection/replay

2.2.1 Unreliable ESs (Internal Adversaries)

In the EDIV process, ESs may exhibit dishonest behavior by falsely claiming that the received data is intact for their own benefit. Such internal attacks of ESs compromise the trustworthiness of the verification process and may lead to corrupted data replicas being mistakenly considered intact, ultimately weakening the integrity of the entire system. Specifically, unreliable ESs may launch the following attacks [11], [15]:
Replay Attacks: An ES may reuse previously valid integrity proofs to deceive the AV into believing that outdated or modified data replicas are still intact. This leads to false verification results and weakens the integrity guarantees.
Replace Attacks: In this case, a dishonest ES intercepts the AV's challenge and responds by substituting genuine data or proofs with unauthorized or tampered ones, thus misleading the AV.
Forgery Attacks: An ES can generate forged integrity proofs for compromised data and present them as valid during the verification process, deceiving the AV and violating the trust of the data.

2.2.2 Unreliable Transmission Channels (External Adversaries)

During the transmission of integrity proofs, the communication channel (backhaul network) may be unreliable and susceptible to external attacks [17]. Attackers may intercept, manipulate, or disrupt integrity proofs while in transit, leading to invalid verification results and compromised security. This dual-layer adversary model, covering both dishonest edge servers and on-path network manipulators, extends prior EDIV formulations that typically assume secure transport channels, enabling end-to-end integrity verification across untrusted networks. It is noted that deployment scenarios involving aggressive packet normalization, encrypted transport protocols that hide TCP-visible carriers, or more complex protocol-layer interference are outside the current scope of this work and would be investigated as future directions.

2.3 Design Goal

Given the outlined threat model, our proposed RAMTStego-EDIV approach is designed to meet the following objectives.
Robustness: Verify the integrity of data replicas even under localized tampering by maximizing hash interdependence across the RAMT structure (strong tamper propagation), and detect in-transit manipulations via steganography-

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

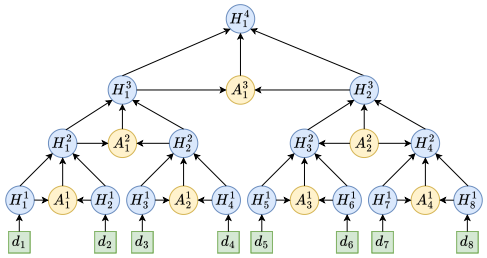


Fig. 2: Illustrative example of the RAMT structure constructed from eight data blocks, showing the formation of auxiliary nodes and ternary parent-hash aggregation.

based proof transmission over the backhaul.

Security: Thwart dishonest ESs attempting replay, replacement, or forgery within the challenge–response protocol, and resist on-path adversaries that alter or replay proofs on untrusted backhaul links.

3 The Proposed RAMTStego-EDIV Framework

This section details the rationale for the design and structural components of the proposed RAMTStego-EDIV framework to address the vulnerabilities inherent in traditional EDIV methods, especially within untrusted network environments.

3.1 Framework Overview

RAMTStego-EDIV is composed of two components, each designed to address two aspects of the resilient and secure EDIV process.

Robust Proof Generation: Generate robust integrity proofs using an enhanced MHT variant called RAMT, which provides secure validation of cached-data replicas.

Dual-Parity TCP Steganography-Based Proof Transmission: Implement a steganographic approach that utilizes dual-parity TCP, enabling covert transmission of the generated integrity proof, specifically the RAMT root, across the untrusted network.

3.2 RAMT-Based Proof Generation

The first component of RAMTStego-EDIV employs RAMT, an improved MHT structure designed to enhance the robustness of integrity proofs. Enhancements and variants of the Merkle Hash Tree (MHT) remain an active and legitimate line of research across cloud, edge, and blockchain domains, which motivates ongoing refinements that strengthen robustness and verifiability. Nevertheless, the design goal of RAMT differs from that of most existing MHT refinements. Rather than primarily pursuing proof-size reduction or hashing efficiency, RAMT is introduced to strengthen tamper-propagation robustness under localized corruption. Unlike traditional binary MHT, RAMT utilizes a ternary aggregation approach at each internal node to strengthen dependencies among nodes, significantly reducing susceptibility to localized data tampering or forgery. To bind the proof structure to the current verification session while preserving consistent reconstruction at both the AV and ES, RAMT applies a deterministic shuffle of the leaf order based on the challenge-specific random seed.

Technically, the RAMT is a binary-structured MHT that incorporates a ternary aggregation mechanism to enhance the robustness of the proof. In a RAMT, each non-leaf node is constructed by hashing three components: a real node, an auxiliary node (derived from adjacent real nodes), and another real node. The overall structure remains binary, maintaining a two-child linkage at each level, but the intermediate node generation leverages ternary aggregation to strengthen inter-node dependencies. A RAMT possesses the following structural properties, while the full RAMT structure is illustrated in Fig. 2 and Algorithm 1 formally describes the RAMT-based proof generation process.

Algorithm 1 Construction of the RAMT

Require: Dataset $D = \{d_1, d_2, \ldots, d_n\}$, random seed rs
Ensure: Merkle root $RAMT_{\text{Root}}$
procedure ComputeRAMT(D, rs)
  function GenerateParentNodes($\mathcal{N}$)
    if $|\mathcal{N}| \bmod 2 \neq 0$ **then**
      Duplicate last node: $H_{\text{dup}} \leftarrow \mathcal{N}[\text{last}]$
      Append H_{dup} to $\mathcal{N}$
    end if
    Initialize: $\mathcal{N}' \leftarrow \emptyset$
    for each consecutive pair $(H_a, H_b) \in \mathcal{N}$ **do**
      $A_{ab} \leftarrow \text{Hash}(H_a \| H_b)$ ▷ Auxiliary hash
      $H_{\text{parent}} \leftarrow \text{Hash}(H_a \| A_{ab} \| H_b)$
      Append H_{parent} to $\mathcal{N}'$
    end for
    return $\mathcal{N}'$
  end function
  Randomized Leaf Hashing with Seed rs
  Let $\pi \leftarrow \text{DeterministicShuffle}([1, 2, \ldots, n], rs)$
  Initialize the leaf node list: $\mathcal{N} \leftarrow \emptyset$
  for each $i \in \pi$ **do** ▷ Randomized index order
    $H_i^1 \leftarrow \text{Hash}(d_i)$
    Append H_i^1 to $\mathcal{N}$
  end for
  while $|\mathcal{N}| > 1$ **do**
    $\mathcal{N} \leftarrow \text{GenerateParentNodes}(\mathcal{N})$
  end while
  return Final Merkle root $RAMT_{\text{Root}} \leftarrow \mathcal{N}[0]$
end procedure

- RAMT is organized as a binary tree, where each non-leaf node has exactly two child nodes. However, each internal node is computed by aggregating three elements: a real node, an auxiliary node (derived from two adjacent nodes at the lower level), and another real node. This ternary aggregation strengthens the dependency between neighboring data blocks. Additionally, if the number of nodes at any level is odd, the last node is duplicated before auxiliary node creation to ensure consistent ternary aggregation. This duplication maintains structural balance and ensures that every aggregation step combines exactly three elements.
- Every node in a RAMT is associated with two identifiers indicating its level l and position i within that level. The level l starts at 1 at the root and increases from top to bottom. Nodes at each level are indexed sequentially from left to right. A node at level l and position i is denoted as H_i^l.
- For every two consecutive real nodes H_{2i-1}^l and H_{2i}^l at a given level l, an auxiliary node A_i^l is generated as:
$$A_i^l = \text{Hash}(H_{2i-1}^l \| H_{2i}^l). \quad (1)$$
This auxiliary node assists in ternary aggregation during the construction of parent nodes at level l.
- Each parent node H_i^l at level l is computed as the cryptographic hash of three inputs:
$$H_i^l = \text{Hash}(H_{2i-1}^{l-1} \| A_i^{l-1} \| H_{2i}^{l-1}). \quad (2)$$

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

This ternary aggregation approach recursively continues across higher layers. If an odd number of nodes remain at any aggregation level, the last node is duplicated to maintain structural consistency. Finally, at the top layer, two intermediate nodes are aggregated along with a final auxiliary node to compute the RAMT root.

- The RAMT structure enhances integrity robustness by introducing auxiliary cross-dependencies between sibling nodes, causing any single modified block to influence a larger portion of the tree. This "tamper-amplification" effect improves detection sensitivity under partial or budget-limited audits while retaining the logarithmic proof size and efficiency of standard Merkle Trees. Importantly, the auxiliary nodes strengthen internal dependency during proof construction but do not enlarge the transmitted authentication path, because the verifier recomputes each auxiliary value locally from the path node and its sibling at the corresponding level. Unlike prior variants that focus mainly on proof-size reduction or hashing optimization, RAMT treats detection robustness as a primary design objective. As formalized in Theorem 3, this structure provides stronger resilience to selective or localized tampering compared to conventional MHTs.

3.3 Steganography-Based Proof Transmission

This second component of RAMTStego-EDIV securely transmits the RAMT-generated integrity proofs from the ES to the AV over potentially adversarial networks. Instead of explicitly transmitting these proofs, which could risk interception or manipulation, RAMTStego-EDIV covertly embeds the proof bits by distributing them across two TCP header fields: the parity of TCP payload length increments (*SNP*) and the least significant bit of the TCP window size (*WSP*). By jointly embedding into both fields, the method doubles the embedding capacity and enhances resilience against detection by network middleboxes. Importantly, the scheme avoids direct manipulation of fragile transport fields such as raw TCP sequence numbers. Instead, it relies only on minimal payload-length parity adjustment and a one-bit modification to the advertised TCP window field, both of which remain aligned with standard TCP behavior across heterogeneous endpoints. Specifically, the RAMT root $RAMT_{\text{Root}}$ is partitioned into binary segments $\{s_1, s_2, \ldots, s_{2m}\}$, with each segment being embedded alternately across the SNP (payload length parity) and WSP fields, maximizing stealth and robustness.

3.3.1 Embedding Procedure

Let the binary representation of the RAMT root be:

$$RAMT_{root} \rightarrow \{s_1, s_2, s_3, \ldots, s_{2m}\}$$

These bits are partitioned into two interleaved subsequences as, $S^{snp} = \{s_1, s_3, s_5, \ldots, s_{2m-1}\}, S^{wsp} = \{s_2, s_4, s_6, \ldots, s_{2m}\}$. We transmit m packets; in each packet i, we embed one SNP bit and one WSP bit. Let SEQ_i, WND_i, and PL_i represent the original TCP sequence number, TCP window size, and TCP payload length, respectively. The adjusted values SEQ'_i, WND'_i, and PL'_i are computed as follows.

Embedding SNP bit s_{2i-1} via Payload Length Parity: Instead of modifying the sequence number directly, which would disrupt TCP transmission, we minimally adjust the payload length by appending one-byte padding (if necessary) to achieve the desired parity:

$$PL'_i = \begin{cases} PL_i, & \text{if } PL_i \mod 2 = s_{2i-1}, \\ PL_i + 1, & \text{otherwise.} \end{cases} \tag{3}$$

TCP automatically computes and manages the sequence number increment based on the adjusted payload lengths as follows:

$$SEQ'_i = SEQ'_{i-1} + PL'_{i-1} \tag{4}$$

Thus, the sequence number is not manually over-written by the embedder; rather, the desired parity signal is induced indirectly through a minimal payload adjustment that remains compatible with standard TCP processing. **Embedding WSP bit s_{2i}:** The WSP embedding remains straightforward. We minimally adjust the advertised window size as follows:

$$WND'_i = \begin{cases} WND_i, & \text{if } WND_i \mod 2 = s_{2i}, \\ WND_i + 1, & \text{otherwise.} \end{cases} \tag{5}$$

Since this modification affects only the least significant bit of the window field, it introduces negligible operational disturbance and remains consistent with the dynamically varying nature of advertised TCP window values in practical deployments. Hence, each packet carries two proof bits, one embedded via payload-length parity and the other via WSP. The embedding adjustments are minimal, conform naturally to TCP semantics, and typically indistinguishable from ordinary traffic behavior.

3.3.2 Extraction Procedure

At the AV side, each embedded bit is extracted simply by inspecting the payload length increment (sequence number difference) and the least significant bit of the window size:

$$s_{2i-1} = PL'_i \mod 2, \quad s_{2i} = WND'_i \mod 2. \tag{6}$$

Therefore, RAMTStego-EDIV explicitly avoids manual adjustments of TCP sequence numbers. All changes are strictly limited to minimal payload padding for SNP signaling and minor window-size adjustment for WSP signaling. This design preserves compliance with TCP's reliability mechanisms and reduces the risk of packet drops or retransmissions caused by abnormal header manipulation. The proposed covert channel is therefore designed for TCP-visible environments in which payload-length parity and TCP window semantics remain observable and usable. Accordingly, transport settings that hide, transform, or remove these carriers, such as encrypted transport protocols, QUIC-based deployments, or aggressively-normalizing middleboxes, fall outside the direct applicability of the present design. The present design is robust to bounded carrier corruption or loss, but systematic over-writing of these fields by aggressive NATs, firewalls, or normalization middleboxes may weaken or disable the covert channel and therefore remains outside the validated deployment scope of this work. Any malicious alteration of these fields either produces detectable errors or causes verification failure, ensuring correctness independent of concealment. An illustrative example of embedding and extraction of proof bits is provided in Appendix A of the supplementary file.

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

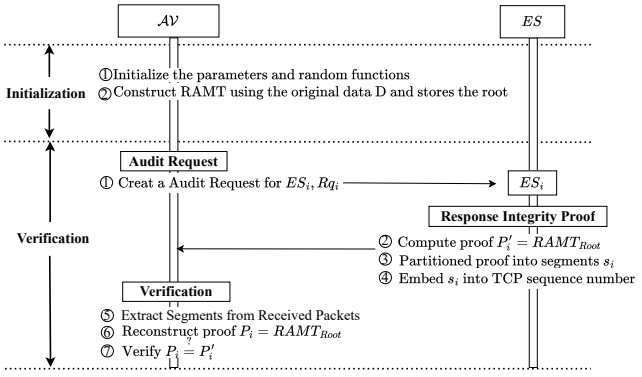


Fig. 3: Workflow diagram of the proposed RAMTStego-EDIV framework.

Algorithm 2 RAMTStego-EDIV Embedding, Extraction, and Verification

Require: Dataset D, Packet stream $\{p_1, p_2, \ldots, p_m\}$, Random seed rs
Ensure: Integrity verification result: {*Intact, Corrupted*}
procedure RAMTSTEGO-EDIV($D, \{p_i\}_{i=1}^m, rs$)
// Phase 1: Initialization
  Initialize system parameters and random functions
  $P_i = \text{COMPUTERAMT}(D, rs)$ ▷ Compute the RAMT root P_i
// Phase 2: Verification
  AV creates a verification challenge to ES_i, chl_i
// Proof Generated by ES_i and send back to AV
  Compute RAMT root $P_i' = \text{COMPUTERAMT}(D, rs)$
  Convert P_i' to binary proof bits $\{s_1, s_2, \ldots, s_{2m}\}$
  Split bits into:$S^{\text{snp}} = \{s_1, s_3, \ldots, s_{2m-1}\}$, $S^{\text{wsp}} = \{s_2, s_4, \ldots, s_{2m}\}$
  for $i \leftarrow 1$ to m **do**
    Get original payload PL_i and window size WND_i
    ▷ SNP embedding: Adjust payload length parity
    if $|PL_i| \bmod 2 \neq s_{2i-1}$ **then**
      Append 1-byte padding marker to PL_i
    end if
    ▷ WSP embedding: Adjust WSP
    if $WND_i \bmod 2 \neq s_{2i}$ **then**
      $WND_i \leftarrow WND_i + 1$
    end if
    Send TCP packet p_i with adjusted PL_i and WND_i
  end for
// Proof Extraction and Verification by AV
  Initialize $Extbit \leftarrow \emptyset$, $PrSEQ \leftarrow InitialSEQ$
  for $i \leftarrow 1$ to m **do**
    Compute $PayloadLen_i' \leftarrow p_i.SEQ - PrSEQ$
    Extract SNP bit: $s_{2i-1}' \leftarrow PL_i' \bmod 2$
    Extract WSP bit: $s_{2i}' \leftarrow p_i.WND \bmod 2$
    Append s_{2i-1}', s_{2i}' to $Extbit$
    $PrSEQ \leftarrow p_i.SEQ$ ▷ Reconstruct original payload
    if last byte of $p_i.PL$ is padding marker **then**
      Remove last byte from $p_i.PL$
    end if
  end for
  $P_i' \leftarrow \text{RECONSTRUCT}(Extbit)$ ▷ Reconstruct proof
  if $P_i' = \text{COMPUTERAMT}(D, rs)$ **then**
    Output: *Intact*
  else
    Output: *Corrupted*
  end if
end procedure

3.4 RAMTStego-EDIV Workflow

The RAMTStego-EDIV framework enables robust and covert EDIV through a novel integration of the RAMT-based proof generation and the TCP-layer steganographic processes. The complete workflow consists of two key stages: initialization and verification. The details of the RAMTStego-EDIV workflow are summarized in Algorithm 2, and depicted in Fig. 3.

3.4.1 Initialization Stage

The AV constructs a RAMT using the cached data and retains its root as the integrity proof for future verification. Several critical parameters are established in this stage. To defend against replay and forgery attacks (as discussed in Section II.B), AVs use a secure random function to generate a challenge-specific random seed rs. This seed ensures that the leaf ordering used for proof construction is session-specific, thereby preventing the reuse of proofs and bolstering replay-attack resistance. Since the permutation is deterministic for a given seed, both the AV and the ES can reconstruct the same leaf order consistently, which is necessary for correct proof generation and verification. The parameter rs is shared between the AV and ESs during each verification round. However, to ensure freshness and unpredictability, a new rs must be generated for each new verification session, distinct from its predecessors.

3.4.2 Verification Stage

This stage involves challenge generation, proof construction, steganographic embedding and transmission, and final verification. The process begins when the AV issues a challenge Chl_i to a target edge server ES_i. This challenge is derived using a time-sensitive function and incorporates a newly generated seed rs_i, which guarantees that even repeated queries over identical data result in distinct proofs, enhancing resistance to both replay and pre-computation attacks. The challenge Chl_i is then securely transmitted to the designated ES. Upon receiving the challenge, ES_i locally constructs a RAMT using its cached dataset D and the received seed rs_i, expressed as $P_i = \text{COMPUTERAMT}(D, rs_i)$. The output is a 256-bit root hash representing a compact and tamper-evident proof of data integrity. Unlike traditional Merkle Trees, the RAMT structure supports dynamic balancing and is designed to accommodate heterogeneous numbers of data replicas, making it particularly suited for distributed and resource-constrained edge environments.

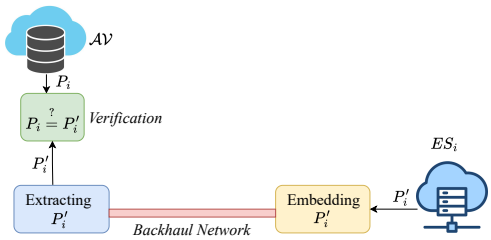


Fig. 4: EDIV process of RAMTStego-EDIV.

Rather than transmitting the proof explicitly—thereby exposing it to interception or tampering—RAMTStego-EDIV employs a steganographic strategy to covertly embed the proof within TCP packet flows. The proof transmission and verification process is illustrated in Fig. 4. The 256-bit proof is serialized into a bitstream $\{s_1, s_2, \ldots, s_{256}\}$ and transmitted using two covert TCP header channels. The odd-indexed bits $\{s_1, s_3, \ldots\}$ are embedded via the SNP channel by adjusting packet padding so that the resulting payload length reflects the desired parity (even or odd). Meanwhile, the even-indexed bits $\{s_2, s_4, \ldots\}$ are embedded through the WSP channel by modifying the least sig-

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

nificant bit (LSB) of the TCP window size field. Each TCP packet thus carries two covert bits, allowing the complete 256-bit proof to be transmitted over just 128 modified packets. These packets conform to normal traffic patterns and appear benign to external observers.

To recover the proof, the AV passively monitors the incoming TCP stream and performs bit extraction from both covert channels. The SNP bits are extracted by analyzing the parity of the payload lengths, and the WSP bits are extracted by reading the LSBs of the TCP window size fields. Once all bits have been collected and reassembled in the correct sequence, the AV reconstructs the proof P_i' and compares it with the reference RAMT root stored during initialization. If the values match, the AV concludes that the data at ES_i has remained intact. However, a mismatch indicates possible data tampering on the server or manipulation during transmission, in which case the AV flags the ES for further scrutiny. Through this seamless and covert verification protocol, RAMTStego-EDIV achieves both *proof confidentiality* and *communication stealth*, ensuring that integrity checks remain hidden from adversaries. Its architecture is inherently robust to both internal threats (e.g., replay or substitution attacks) and external threats (e.g., packet inspection or interference). Furthermore, its lightweight computational and communication overhead, coupled with cryptographic soundness and adaptability to heterogeneous edge environments, makes RAMTStego-EDIV a practical and resilient solution for the verification of secure edge data on untrusted or adversarial networks.

4 Theoretical Results Analysis

This section provides the theoretical analysis that includes the correctness, efficiency, and security analysis of the proposed RAMTStego-EDIV approach. The goal is to formally establish the feasibility, optimality, and robustness of our method under adversarial edge environments.

Lemma 1 (Correctness)**.** *If a data block d_j within a cached replica is modified, the RAMT-based proof generation process will yield a root hash $H_1^{l\,\prime}$ that differs from the original H_1^l provided that the employed Hash() function is collision-resistant.*

The detailed proof for Lemma 1 is provided in Appendix B of the supplementary document. It illustrates that RAMTStego-EDIV accomplishes the goal outlined in Section II.C, namely, the objective of correctness.

Theorem 1 (Correctness)**.** *Let the RAMT root hash $h = RAMT_{Root} \in \{0,1\}^k$ be partitioned into $m = \lceil k/2 \rceil$ consecutive two-bit segments $\mathbf{s} = \{s_1, s_2, \ldots, s_m\}$, where each segment $s_i = (s_i^{(1)}, s_i^{(2)}) \in \{0,1\}^2$. Suppose each segment s_i is embedded into a TCP packet P_i by encoding $s_i^{(1)}$ as the parity of the sequence number (SNP) and $s_i^{(2)}$ as the least significant bit of the window size (WSP). If all packets $\{P_1, \ldots, P_m\}$ are received without modification to these two header fields, then the receiver can deterministically reconstruct the exact value of h. That is,*

$$\hat{h} = \texttt{Extract}(P_1, \ldots, P_m) = RAMT_{Root}.$$

The detailed proof for Theorem 1 is provided in Appendix B of the supplemental document. It illustrates that RAMTStego-EDIV accomplishes the goal outlined in Section II.C, namely, the objective of correctness.

Theorem 2 (Robustness)**.** *Let $D = \{d_1, \ldots, d_n\}$ be a dataset. Let $\mathcal{T}_{mht}$ and $\mathcal{T}_{ramt}$ be Merkle trees over D, constructed using binary MHT and RAMT, respectively. For any tampered leaf d_i, let $\mathcal{D}_{mht}(d_i) \subseteq \mathcal{T}_{mht}$ and $\mathcal{D}_{ramt}(d_i) \subseteq \mathcal{T}_{ramt}$ denote the sets of internal nodes whose values are affected by the modification of d_i. Then,*

$$|\mathcal{D}_{ramt}(d_i)| > |\mathcal{D}_{mht}(d_i)|.$$

The detailed proof for Theorem 2 is provided in Appendix B of the supplemental document. It illustrates that RAMTStego-EDIV accomplishes the goal outlined in Section II.C, namely, the objective of robustness.

Theorem 3 (Tamper-Propagation Amplification of RAMT)**.** *Let $D = \{d_1, \ldots, d_n\}$ with $n \geq 4$. Consider (i) a b-ary Merkle tree T_b over D for fixed $b \geq 2$ (with standard padding) and (ii) the RAMT T_{ramt} from equation (1) and (2). Count as* internal *all non-leaf nodes (parents and auxiliaries). For a uniformly random leaf index J, let $\Delta_b(J)$ and $\Delta_{\text{ramt}}(J)$ be the number of internal nodes whose hash values change after replacing d_J by $d'_J \neq d_J$. Under equal proof length (auxiliaries recomputable, not transmitted), we have*

$$\begin{aligned} \mathbb{E}[\Delta_{\text{ramt}}(J)] &> \mathbb{E}[\Delta_b(J)], \\ \mathbb{E}[\Delta_{\text{ramt}}(J)] &\geq \lceil \log_2 n \rceil + 1, \\ \mathbb{E}[\Delta_b(J)] &= \lceil \log_b n \rceil. \end{aligned} \tag{7}$$

The detailed proof for Theorem 3 is provided in Appendix B of the supplemental document. It illustrates that RAMTStego-EDIV accomplishes the goal outlined in Section 2.3, namely, the objective of robustness.

Theorem 4 (Security)**.** *Let D be a dataset of the AV and cached at an ES. Let $R = \mathsf{RAMT}(D)$ be the root of a RAMT computed using a collision-resistant hash function H, and let $R = r_1, \ldots, r_k \in \{0,1\}^k$ denote its bitwise representation. Suppose the bits r_i are embedded and transmitted over a TCP flow using the proposed steganography. Then, under the assumption that H is collision-resistant and the embedded bits are uniformly random, the RAMTStego-EDIV protocol is secure against different types of internal attacks by untrusted ES and external attacks by the untrusted backhaul network.*

The detailed proof for Theorem 3 is provided in Appendix B of the supplemental document. It illustrates that RAMTStego-EDIV accomplishes the goal outlined in Section II.C, namely, the objective of security.

Theorem 5 (Embedding Cost of RAMTStego-EDIV)**.** *Let $R \in \{0,1\}^\lambda$ be the RAMT root generated over a dataset D, and let $R = r_1, r_2, \ldots, r_\lambda$ denote its binary representation. In the RAMTStego-EDIV protocol, where these bits are embedded into TCP packet headers using parity-based fields, the total computation cost of embedding is $O(\lambda)$.*

The detailed proof for Theorem 4 is provided in Appendix B of the supplemental document. It illustrates the embedding cost of the proposed RAMTStego-EDIV.

Theorem 6 (Dual-Carrier Coded Covert Channel: Reliability and Indistinguishability)**.** *Let $R \in \{0,1\}^\lambda$ be encoded by a binary linear code C with blocklength n_c, dimension k_c, and error/erasure radius t. Interleave the n_c coded bits across*

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

$m = \lceil n_c/2 \rceil$ TCP packets using two carriers per packet: (i) SNP—payload-length parity via at most $+1$ byte pad; (ii) WSP—window-size LSB via at most $+1$ increment. Let the embedder use Bernoulli smoothing with parameter $\rho \in (0, 1)$.

(Reliability) *If per codeword the adversary corrupts at most t carrier symbols (bit flips or erasures), decoding succeeds with probability 1. Under i.i.d. carrier loss with probability $p < t/n_c$,*

$$\Pr[\text{decode fail}] \;\le\; \exp\Big(-\, n_c\, D\big(\tfrac{t}{n_c} \,\|\, p\big)\Big),$$

where $D(a\|p) = a \ln \frac{a}{p} + (1-a) \ln \frac{1-a}{1-p}$ is the binary KL divergence.

(Indistinguishability) *Let $(X, Y) \in \{0, 1\}^2$ denote the clean joint law of (payload parity, window LSB) and let U be uniform on $\{0, 1\}^2$. If $\delta_0 = \mathrm{TV}((X, Y), U)$, then the post-embedding pair $(\tilde{X}, \tilde{Y})$ satisfies*

$$\mathrm{TV}\big((\tilde{X}, \tilde{Y}), U\big) \;\le\; \delta_0 + \rho.$$

Hence, no detector can distinguish embedded from clean traffic with an advantage exceeding $\delta_0 + \rho$.

The detailed proof for Theorem 6 is provided in Appendix B of the supplemental document. It demonstrates that the dual-carrier covert channel achieves the extended objectives of reliability and indistinguishability.

5 Experimental Results Analysis

In this section, we present comprehensive experiments under controlled adversarial settings to evaluate the security, robustness, and efficiency of the proposed RAMTStego-EDIV framework, particularly by comparing its performance with established EDIV solutions.

5.1 Experimental Settings

We present comprehensive experimental results to evaluate the effectiveness of our proposed RAMTStego-EDIV scheme. Each experiment is repeated 50 times, and the results are averaged unless otherwise stated. All experiments are conducted using Python 3.11 on a computing system equipped with an Intel Core i5 processor (1.3 GHz) and 16 GB of RAM. While the main evaluation emphasizes mechanism-level scalability and robustness under controlled experimental settings, the supplementary document additionally includes a real-world device-to-device transmission study to assess the practical feasibility of the TCP-based covert channel over a real TCP network stack. Specifically, the proof-transmission mechanism is validated across two heterogeneous physical machines connected through a local Wi-Fi network, thereby providing complementary evidence beyond abstract simulation.

5.1.1 Baseline Approach

To evaluate the effectiveness of the proposed RAMTStego-EDIV framework, we compare its performance against existing state-of-the-art EDIV methods. Specifically, three representative baseline schemes are considered: EDI-V [15], EDI-S [11], and ICL-EDI [7]. These methods are selected due to their widespread adoption in the literature and their use of MHT-based or cryptographic verification mechanisms.

TABLE 2: Parameter settings

Parameters	Value Varied	Value Fixed
Edge Scale (n)	10, 20, 30, 40, 50, 60	50
Data Scale (k)	16, 32, 64, 128, 256, 512	256
Data Size (ds)	16, 32, 64, 128, 256, 512, 1024	256
Corrupted data Ratio (Cr)	0.01, 0.02, 0.03, 0.04, 0.05	0.03
Proof tampering Ratio (ptr)	0.01, 0.02, 0.03, 0.04, 0.05	0.04

5.1.2 Performance Evaluation Metric

To assess the effectiveness of the proposed RAMTStego-EDIV framework, we consider both conventional EDIV metrics and robustness-oriented metrics. The conventional EDIV metrics include computation cost and communication cost, which evaluate processing efficiency and communication overhead, respectively. In addition, we use precision and Stealth Transmission Integrity (STI) to assess resilience against data tampering and transmission-side interference. Precision measures whether corrupted and intact states are correctly identified during verification. In contrast, STI evaluates the reliability of the covert proof-transmission channel itself. Specifically, STI is defined as the proportion of covertly transmitted integrity proofs that are successfully reconstructed at the verifier without corruption or disruption in transit:

$$STI = \frac{N_{\text{succ}}}{N_{\text{tx}}} \times 100\%, \tag{8}$$

where N_{tx} denotes the total number of transmitted covert proofs and N_{succ} denotes the number of proofs that are correctly recovered and reconstructed by the AV. Thus, STI captures transmission robustness rather than the classification correctness of the EDIV decision.

5.1.3 Parameter Settings

To comprehensively assess the efficacy of RAMTStego-EDIV, we define the evaluation parameters listed in Table 1, following the conventions established in previous EDIV studies [7], [9], [11]. Edge Scale (n) denotes the total number of ESs participating in the verification process, which varies between 10 and 60. Data Scale (k) indicates the total number of data blocks maintained by the AV and distributed across ESs. Data Size (ds) represents the size of each data replica, measured in kilobytes. Corrupted Block Ratio (cr) specifies the proportion of corrupted blocks within each compromised replica, defined as the ratio of tampered blocks to the total number of blocks. The selected cr range ($1\%-5\%$) reflects low-to-moderate localized corruption, which is relevant in MEC settings where attacks may be sparse and selective. Proof Tampering Ratio (ptr) denotes the proportion of integrity proofs that are tampered with during transit. Likewise, the selected ptr range (1%–5%) represents moderate backhaul interference and is used to evaluate the resilience of the covert proof-transmission mechanism under increasing levels of in-transit manipulation. These parameter ranges are intended as representative stress settings for the two vulnerabilities studied in this work, rather than as claims of exact prevalence across all MEC deployments. For each evaluation, a single parameter is adjusted while the remaining parameters are maintained at their default values. The experiments are systematically repeated, and the results are recorded to derive the mean values, ensuring a robust and thorough analysis [7], [11].

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

5.2 Experimental Results Related to Robustness

To evaluate system robustness, we assess precision, analyze the framework's resilience to internal threats, and examine the tamper-propagation characteristics of RAMT (see **Appendix C.A** in the supplementary document). We also measure the Stealth Transmission Integrity to quantify the resilience of the dual-parity covert channel.

5.2.1 Precision

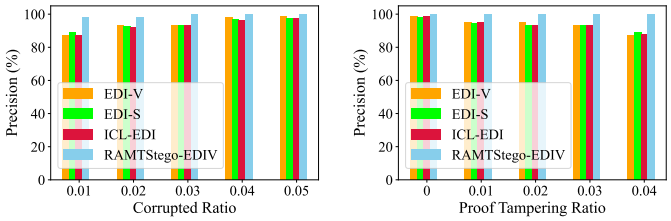
We conduct a comprehensive analysis to evaluate the precision of the proposed RAMTStego-EDIV, examining the factors that influence the precision with the baseline methods.

- Impact of cr: To assess the effectiveness of each method in accurately detecting corrupted data blocks, we evaluate the precision across varying levels of cr (1% to 5%). As depicted in Fig. 5(a), all methods demonstrate a rising trend in precision as the corruption ratio increases. This suggests that the detection mechanisms become more effective when a higher proportion of blocks are genuinely corrupted. Among the compared techniques, *RAMTStego-EDIV* consistently achieves the highest precision, maintaining near-perfect accuracy even at low corruption levels and reaching 100% precision at a 5% corruption ratio. This highlights its strong ability to avoid false positives while reliably detecting actual integrity violations. In contrast, while *EDI-V*, *EDI-S*, and *ICL-EDI* also show improvements with increased corruption density, their precision is relatively lower in the 1–2% range, indicating reduced reliability under sparse corruption. These results suggest that RAMTStego-EDIV offers superior trustworthiness and precision, particularly in environments with low but critical corruption.

- Impact of ptr: To examine the robustness of each method under adversarial transmission conditions, we vary the ptr to simulate the deliberate corruption of integrity proofs during transit. As depicted in Fig. 5(b), the precision of conventional methods such as *EDI-V*, *EDI-S*, and *ICL-EDI* declines progressively as ptr increases from 0% to 4%, reflecting their sensitivity to tampered or malformed proof messages. Such degradation likely results from increased false positives due to mismatched or unverifiable proofs. In contrast, *RAMTStego-EDIV* maintains 100% precision across all tested ptr levels, demonstrating strong resilience against transmission-layer manipulation. This robustness is attributed to its steganography-based proof embedding within TCP headers, which enhances resistance to tampering and supports consistent verification. These findings confirm that RAMTStego-EDIV not only performs well under ideal conditions but also sustains high precision in adversarial communication environments, making it a compelling choice for integrity-critical edge applications.

5.2.2 Robustness Against Internal Threats

To assess the robustness of RAMTStego-EDIV against internal adversaries, we conducted a controlled evaluation involving three common internal threat scenarios: replay attacks, proof replacement, and forgery attempts. As shown in Table 3, each attack type was executed in 50 independent trials, totaling 150 adversarial attempts. In all cases, RAMTStego-EDIV achieved a 100% detection success rate, indicating its capability to consistently recognize malicious



(a) Impact of cr (b) Impact of ptr

Fig. 5: Precision analysis.

TABLE 3: Evaluation of RAMTStego-EDIV security against the internal threats

Attack Type	No. of Trials	Detection Success Rate	False Positive Rate	Hash Collisions Observed
Replay	50	100%	0%	0
Replace	50	100%	0%	0
Forgery	50	100%	0%	0
Overall	150	100%	0%	0

or reused integrity proofs. Moreover, the system exhibited a 0% false positive rate, confirming that no legitimate proofs were mistakenly flagged as tampered with. Importantly, no hash collisions were observed across all trials, further validating the cryptographic soundness of the Merkle-based proof structure embedded via TCP steganography. These results demonstrate that RAMTStego-EDIV is highly resilient to common internal manipulation strategies, making it a strong candidate for secure and trustworthy edge data verification in adversarial environments.

5.2.3 Stealth Transmission Integrity

To evaluate the resilience of covert integrity proof delivery, we measured the STI of RAMTStego-EDIV and baseline methods under varying levels of ptr. Following the definition in Section 6.1.2, STI denotes the percentage of transmitted proofs that are successfully reconstructed at the verifier without corruption or disruption during transit. Therefore, a higher STI indicates that the proof-transmission mechanism is more robust to in-transit tampering and packet-level interference. As shown in Fig. 6, RAMTStego-EDIV consistently maintains near-perfect STI, starting at 100% when ptr is 0% and only slightly declining to 98.7% at ptr = 4%, demonstrating its high resilience to transmission-layer tampering. In contrast, the baseline methods show a sharper degradation. Specifically, EDI-V drops from 100% to 96.5% with just 1% tampering and further declines to 84.3% at ptr = 4%. EDI-S and ICL-EDI follow similar trends, with STI values falling to 82.7% and 86.6%, respectively, under the same conditions. These results highlight RAMTStego-EDIV's robustness in preserving proof integrity despite active tampering attempts, primarily due to its lightweight, distributed steganographic embedding across TCP headers. In contrast, traditional methods that rely on explicit transmission of integrity proofs are more susceptible to disruption, emphasizing the advantage of covert and resilient transmission strategies in adversarial environments. The supplementary document further reports a real device-to-device transmission experiment over two heterogeneous machines connected via Wi-Fi. The proposed scheme achieved 100% proof-bit extraction accuracy with 0% packet loss across 10 independent trials. Although

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

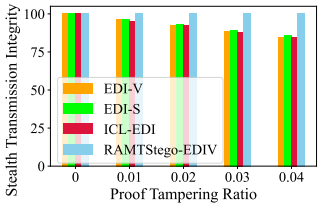


Fig. 6: Stealth transmission integrity.

not a full MEC deployment, this result provides practical evidence that the covert channel operates reliably across heterogeneous endpoints and a real TCP network stack.

5.3 Experimental Results Related to Existing EDIV

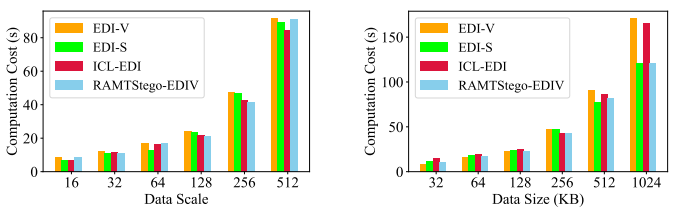
5.3.1 Computation Cost

We conduct a comprehensive analysis to evaluate the time consumption across different stages of the RAMTStego-EDIV process, examining the factors that influence the duration of each stage.

Computation Cost in the Initialization Stage

- Impact of k: To assess the initialization overhead introduced by each method, we measured computation cost in seconds across varying data scales ranging from 16 to 512 data blocks. As shown in Fig. 7(a), all methods exhibit an increasing trend as the number of blocks grows. RAMTStego-EDIV maintains comparable performance with existing approaches, recording 8.3 seconds at 16 blocks and 21.1 seconds at 128 blocks. At the largest scale of 512 blocks, its cost rises to 90.6 seconds, which remains in line with EDI-V (91.35 s) and EDI-S (89.36 s). ICL-EDI demonstrates slightly better efficiency, reaching only 84.6 seconds at the same scale, due to its lightweight initialization logic. While RAMTStego-EDIV introduces moderate additional overhead for proof embedding and Merkle root preparation, it scales linearly and remains practically feasible across data volumes.

- Impact of ds: We further analyzed the initialization cost concerning the increase in data sizes, from 32 KB to 1024 KB, as shown in Fig. 7(b). All methods show consistent cost growth as the data volume increases. RAMTStego-EDIV begins at 10 seconds for 32 KB and reaches 120.8 seconds for 1024 KB, performing slightly better than EDI-V (170.36 s) and closely tracking EDI-S (120.36 s). ICL-EDI maintains the lowest initialization cost (165.5 s at 1024 KB), but the performance gap narrows at larger scales. These results confirm that RAMTStego-EDIV remains efficient during the setup phase while providing enhanced security through its steganographic initialization process.



(a) Impact of k (b) Impact of ds

Fig. 7: Computation cost in the initialization stage.

Computation Cost in the Verification Stage

- Impact of n: To examine how verification cost scales with the number of edge servers, we evaluated each method across increasing edge scales from 10 to 60 nodes. As shown in Fig. 8(a), all methods exhibit a linear increase in computation cost with edge scale. RAMTStego-EDIV incurs slightly higher overhead compared to the baselines, with a peak cost of 678.4 s at 50 nodes and 1298.9 s at 60 nodes. This increase is attributed to the additional processing required for the extraction of embedded proofs. EDI-V and EDI-S follow a similar growth trend, while ICL-EDI demonstrates slightly better performance at moderate scales (e.g., 567.5 s at 50 ES) but converges with other methods at higher scales. Despite the modest increase, RAMTStego-EDIV remains computationally feasible for large-scale edge environments.

- Impact of k: We further analyzed the verification cost related to the number of data blocks. All methods show predictable growth as the data scale increases from 16 to 512 blocks, as shown in Fig. 8(b). RAMTStego-EDIV performs comparably with EDI-V and EDI-S at lower scales and incurs a slightly higher cost at the maximum scale (1401.5 s at 512 blocks). This overhead is expected due to the reconstruction of the Merkle tree by RAMTStego-EDIV from steganographically embedded proofs. ICL-EDI remains the most efficient method across the data scale, but the performance gap narrows as the block count increases.

- Impact of ds: Lastly, we evaluated how varying input data sizes, from 32 KB to 1024 KB, affect computation cost, as shown in Fig. 8(c). RAMTStego-EDIV shows a smooth and consistent growth trend, with costs increasing from 45.7 s at 32 KB to 1298.7 s at 1024 KB. While ICL-EDI performs best across smaller data sizes, RAMTStego-EDIV's performance remains close and competitive. The slightly higher overhead stems from its steganographic payload parsing and verification process but remains within acceptable limits for practical applications.

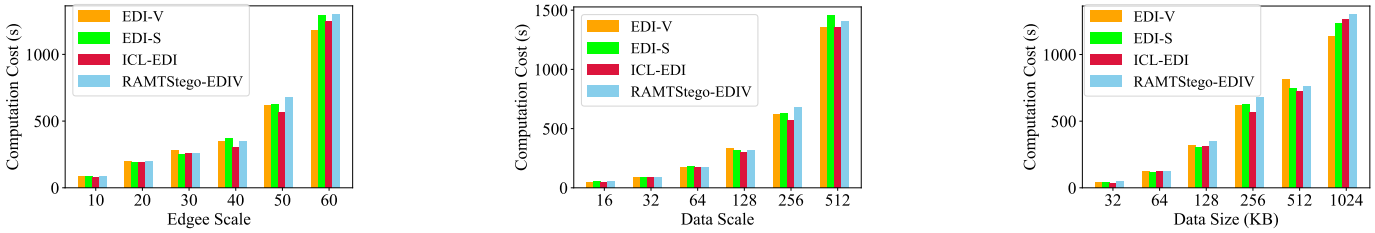
5.3.2 Communication Cost

In RAMTStego-EDIV, integrity proofs are transmitted covertly by embedding the 256-bit RAMT root into standard TCP packets. This is achieved through a dual-parity embedding strategy, where one bit is encoded in the parity of the TCP payload length (SNP) and another in the least significant bit of the TCP window size (WSP), effectively transmitting two bits per packet. As a result, 128 packets are sufficient to transmit a complete 256-bit proof. In the worst case, each packet may require up to two bytes of modification—one for adjusting the payload length parity and another for tuning the window size. This results in a total communication overhead of approximately 256 bytes per proof, which is negligible compared to standard TCP packet sizes (typically around 1,500 bytes). Importantly, this embedding process does not alter the functional behavior of TCP and introduces no statistically detectable patterns, thereby preserving stealth against eavesdropping and manipulation.

Fig. 9 illustrates the communication cost comparison under varying edge server scales and data scales. In Fig. 9(a), as the number of edge servers increases from 10 to 60, existing methods such as EDI-S, EDI-V, and ICL-EDI show a rapid increase in communication cost due to explicit proof transmissions. In contrast, RAMTStego-EDIV maintains a

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006



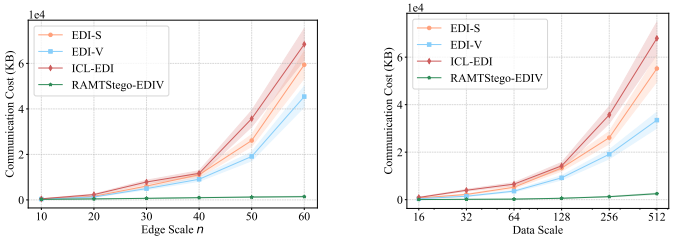
(a) Impact of n (b) Impact of k (c) Impact of ds

Fig. 8: Computation cost in the verification stage.

TABLE 4: Ablation study under $ptr = 3\%$ and $cr = 3\%$.

Config.	Precision (%)	STI (%)	Det. Tamp. (%)
RAMT + Stego Tx (Full)	100.0	99.1	100.0
RAMT + Std. Tx	100.0	72.3	100.0
Binary MHT + Stego Tx	87.4	98.8	87.4
Binary MHT + Std. Tx	87.4	71.8	87.4

nearly constant and minimal overhead regardless of the number of edge servers. Similarly, Fig. 9(b) shows the trend as the data scale increases from 16 to 512. The baseline methods again incur growing communication costs, whereas RAMTStego-EDIV demonstrates excellent scalability with a flat communication footprint. These results confirm that the steganographic transmission scheme of RAMTStego-EDIV is not only lightweight but also inherently scalable with respect to both network and data dimensions.



(a) Impact of n (b) Impact of k

Fig. 9: Communication cost in KB.

5.4 Ablation Study

To isolate the contributions of the proof structure and the transmission mechanism, we evaluated four configurations under a representative adversarial setting with $ptr = 3\%$ and $cr = 3\%$: *(i)* RAMT + Stego Tx (full RAMTStego-EDIV), *(ii)* RAMT + Std. Tx, *(iii)* Binary MHT + Stego Tx, and *(iv)* Binary MHT + Std. Tx. Here, *Stego Tx* refers to the proposed dual-parity covert transmission mechanism, while *Std. Tx* refers to ordinary explicit transmission of the integrity proof without steganographic embedding. The contribution of RAMT is evident by comparing the first and third rows, as well as the second and fourth rows. In both cases, replacing Binary MHT with RAMT improves precision by 12.6 percentage points and increases detected tampering from 87.4% to 100.0%. This confirms that RAMT strengthens proof robustness by amplifying tamper propagation.

The contribution of Stego Tx is observed by comparing the first and second rows, as well as the third and fourth rows. In both cases, replacing standard proof delivery with steganographic transmission substantially improves *STI*. Specifically, STI increases from 72.3% to 99.1% for RAMT-based proofs and from 71.8% to 98.8% for Binary-MHT-based proofs, yielding an average gain of approximately 26.9 percentage points. Most importantly, the full RAMTStego-EDIV configuration achieves the best overall performance, with 100.0% precision, 99.1% STI, and 100.0% detected tampering. This shows that RAMT and Stego Tx are complementary and jointly necessary for end-to-end protection in adversarial MEC environments.

5.5 Extended Experiments

We conducted extended experiments to evaluate the transmission reliability, latency, and computation cost associated with constructing the RAMT, thereby further validating the reliability and timing efficiency of the proposed RAMTStego-EDIV framework. Detailed results are provided in **Appendix C.B to Appendix C.D** of the supplementary material. The findings show that the steganographic transmission scheme consistently delivers integrity proofs under real network conditions without incurring packet loss, reordering, or protocol-level disruptions. The embedded bits were extracted accurately with 100% success in all trials. Additionally, the observed latency and jitter remained within acceptable thresholds, confirming that the embedding process introduces minimal timing overhead and preserves the functional integrity of the TCP protocol. The computation cost results further confirm that RAMT achieves enhanced robustness without compromising efficiency.

6 Related Work

Early research on cloud data integrity (CDI) [20], [21] established foundational schemes such as Proof of Retrievability (PoR) [22], [23] and Provable Data Possession (PDP) [24]. These methods enable an application verifier to verify the integrity of remotely stored files via probabilistic proofs. Despite their influence, many PoR- and PDP-based solutions depend on a trusted third-party auditor (TPA) [25], introducing complexities in trust delegation and security guarantees. Moreover, as edge computing scales to encompass numerous geographically dispersed nodes, directly applying CDI methods becomes inefficient, prompting the need for customized EDI solutions [4].

EDI-specific methods typically follow an interactive

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

TABLE 5: Related work comparison

Scheme	Proof Generation	Transmission Security	Internal Attack	External Attack	Overhead
Tong *et al.* [5]	PDP / PoR	✗	✗	✗	Moderate
Li, *et al.* [15]	MHT-based	✗	✓	✗	Medium
Li *et al.* [11]	ECC-based	✗	✓	✗	High
Li *et al.* [26]	Offloaded to ES	✗	✓	✗	High
Li *et al.* [27]	MHT	✗	✓	✗	High
Cui *et al.* [7]	Homomorphic	✗	✓	✗	Moderate
Ding *et al.* [12]	Dynamic arrays	✗	✓	✗	Moderate
Zhao *et al.* [6]	Smart filtering	✗	✓	✗	Medium
Proposed	RAMT	✓	✓	✓	Low

challenge-response paradigm, where an AV or TPA periodically challenges ESs and collects integrity proofs. Tong et al. [5] were among the first to explore EDI, adapting PDP from the cloud domain and extending its theoretical guarantees in [10], but facing limitations in multi-replica environments [26]. MHT-based schemes, such as EDI-V [15], improve verification efficiency yet still incur significant communication overhead and rely on probabilistic checks. Conversely, Li et al. [11] introduced EDI-S, using elliptic curve cryptography for deterministic guarantees, albeit with increased computation cost. Other approaches, such as CooperEDI [26], offload verification to ESs, reducing AV load but increasing bandwidth usage. EdgeWatch [27] eliminates TPAs by incorporating blockchain incentives but introduces on-chain transaction delays that reduce overall efficiency. Additional schemes address computation and update complexity. Cui et al. [7] presented ICL-EDI, which uses homomorphic signatures for simplified verification, while Ding et al. [12] proposed EDI-DA, a dynamic data verification scheme leveraging index-single linked table structures and dynamic arrays. However, these methods still rely on regular challenge-response interactions, which may not scale well.

To optimize verification in large-scale environments, Zhao et al. [6] proposed a Smart Inspection Algorithm (SIA) to selectively challenge unreliable replicas, later expanding it into a dynamic verification assignment strategy. However, most methods lack an end-to-end solution for replica filtering and scheduling that balances overhead and integrity assurance. Batch verification techniques [10] accelerate proof aggregation but typically require a TPA, raising privacy concerns [28] and potential adversarial exploits [29]. A learning-based method has also been explored for anomaly filtering in [8], yet standard federated aggregation (e.g., FedAvg) remains vulnerable to accuracy loss under data heterogeneity. A summary comparison of these representative methods, including their ability to resist internal and transmission attacks, is presented in Table 5. As shown, while previous work primarily focuses on integrity-proof generation, they neglect secure transmission, leaving EDIV systems vulnerable to man-in-the-middle tampering. In contrast, this is the first transmission-protected EDIV protocol using in-packet steganography to combine a robust Merkle structure with a secure, stealthy transmission protocol designed for harsh network conditions.

7 Conclusion and Future Work

This paper presented RAMTStego-EDIV, a novel two-layer framework designed to enhance the resilience and security of EDIV in adversarial and resource-constrained environments. To ensure robustness against forgery and localized tampering, we introduced the RAMT, a ternary hash structure that combines real and auxiliary nodes to produce tamper-evident integrity proofs. To address vulnerabilities in transmitting these proofs over untrusted networks, we developed a dual-parity steganographic scheme that covertly embeds verification data into TCP protocol fields, ensuring secure and unobtrusive transmission. Our theoretical and experimental analyses confirm that RAMTStego-EDIV effectively mitigates critical threats such as interception, tampering, and replay attacks while maintaining low communication overhead—an essential feature for edge computing applications. The current transmission design is intended for TCP-visible environments, in which the required protocol fields remain observable and usable. By jointly addressing structural and transmission-layer vulnerabilities, the proposed framework offers a practical and scalable solution to ensure the integrity of trustworthy data at the network edge.

Future work will therefore focus on extending the framework beyond the current deployment scope. In particular, one important direction is to design alternative covert transmission mechanisms for settings in which the TCP-visible carriers used in this work are hidden, transformed, or unavailable, such as TLS-protected or QUIC-based environments. Future work will also explore more adaptive steganographic embedding strategies that respond to dynamic network conditions and adversarial behaviors. Extending RAMTStego-EDIV to multi-hop edge architectures and integrating cross-layer anomaly detection are also promising directions for broadening its applicability. Another important direction is to assess robustness under practical network interference such as packet normalization, field rewriting, and traffic manipulation by middleboxes. Finally, future extensions may include designing stronger covert channels with improved statistical indistinguishability, adaptive embedding control, and integrated error-correcting codes to further enhance proof transmission resilience under adversarial conditions.

References

[1] I. Analytics, "Internet of things (iot) and non-iot active device connections worldwide from 2010 to 2025 (in billions)," *Statista*, 2020.
[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, pp. 7–18, 2010.
[3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.
[4] 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.
[5] 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.
[6] Y. Zhao, Y. Qu, F. Chen, Y. Xiang, and L. Gao, "Data integrity verification in mobile edge computing with multi-vendor and multi-server," *IEEE Transactions on Mobile Computing*, 2023.
[7] 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.
[8] Y. Zhao, C. Xu, Y. Qu, Y. Xiang, F. Chen, and L. Gao, "A learning-based hierarchical edge data corruption detection framework in edge intelligence," *IEEE Internet of Things Journal*, 2024.
[9] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, and L. Gao, "An efficient and reliable asynchronous federated learning scheme for smart public transportation," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 6584–6598, 2022.
[10] 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*, 2022.
[11] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Inspecting edge data integrity with aggregate signature in distributed edge computing

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

environment," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2691–2703, 2021.

[12] Y. Ding, Y. Li, W. Yang, and K. Zhang, "Edge data integrity verification scheme supporting data dynamics and batch auditing," *Journal of Systems Architecture*, vol. 128, p. 102560, 2022.

[13] L. Rao, H. Zhang, and T. Tu, "Dynamic outsourced auditing services for cloud storage based on batch-leaves-authenticated merkle hash tree," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 451–463, 2020.

[14] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "Mur-dpa: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.

[15] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Auditing cache data integrity in the edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1210–1223, 2020.

[16] J. Li, Q. Zhao, H. Cheng, S. Teng, N. Wu, and Y. Liang, "Or-edi: 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.

[17] N. Zhang, Y. Zhao, Y. N. Qu, B. Gu, K. Sood, L. Gao, and S. Yu, "From data integrity to global model integrity for federated learning: An mht-based approach," in *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*, 2024, pp. 956–961.

[18] Z. Wu, Z. Xu, and H. Wang, "Whispers in the hyper-space: High-bandwidth and reliable covert channel attacks inside the cloud," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 603–615, 2015.

[19] L. Qiao, Y. Li, F. Wang, and B. Yang, "Lightweight integrity auditing of edge data for distributed edge computing scenarios," *Ad Hoc Networks*, vol. 133, p. 102906, 2022.

[20] F. Zafar, A. Khan, S. U. R. Malik, M. Ahmed, A. Anjum, M. I. Khan, N. Javed, M. Alam, and F. Jamil, "A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends," *Computers & Security*, vol. 65, pp. 29–49, 2017.

[21] L. Zhou, A. Fu, S. Yu, M. Su, and B. Kuang, "Data integrity verification of the outsourced big data in the cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 122, pp. 1–15, 2018.

[22] 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.

[23] C. B. Tan, M. H. A. Hijazi, Y. Lim, and A. Gani, "A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends," *Journal of Network and Computer Applications*, vol. 110, pp. 75–86, 2018.

[24] 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.

[25] W. Guo, H. Zhang, S. Qin, F. Gao, Z. Jin, W. Li, and Q. Wen, "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Generation Computer Systems*, vol. 95, pp. 309–322, 2019.

[26] 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.

[27] B. Li, Q. He, L. Yuan, F. Chen, L. Lyu, and Y. Yang, "Edgewatch: Collaborative investigation of data integrity at the edge based on blockchain," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3208–3218.

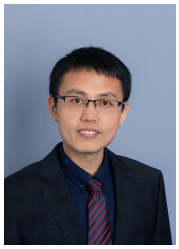
[28] Y. Ren, J. Qi, Y. Liu, J. Wang, and G.-J. Kim, "Integrity verification mechanism of sensor data based on bilinear map accumulator," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–19, 2021.

[29] G. Xu, Y. Yang, C. Yan, and Y. Gan, "A probabilistic verification algorithm against spoofing attacks on remote data storage," *International Journal of High Performance Computing and Networking*, vol. 9, no. 3, pp. 218–229, 2016.

Md Rashedul Islam is currently pursuing a PhD degree at the School of Information Technology, Deakin University. He completed an M.Sc. degree in Computer Science and Engineering at Rajshahi University of Engineering and Technology (RUET), Bangladesh, in 2019. He also received a B.Sc. degree in Computer Science and Engineering from Hajee Mohammad Danesh Science and Technology University (HSTU), Bangladesh. He is an academic faculty member at HSTU, Bangladesh. His research direction includes edge computing, machine learning, FL, blockchain applications, and remote sensing image analysis.

Yong Xiang received the Ph.D. degree in Electrical and Electronic Engineering from The University of Melbourne, Australia. He is a Professor at the School of Information Technology, Deakin University, Australia. His research interests include cybersecurity, data science, machine learning & AI, distributed computing, and communications engineering. He has published 7 authored books, over 300 journal articles, and over 120 conference papers in these areas. Professor Xiang is the Associate Editor of IEEE Communications Surveys and Tutorials, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Cloud Computing, and Computer Standards and Interfaces. He was the Senior Area Editor of IEEE Signal Processing Letters and the Associate Editor of IEEE Access, and served as Guest Editor for several journals, such as IEEE Transactions on Industrial Informatics, IEEE Transactions on Intelligent Transportation Systems, IEEE Multimedia, etc. He has served as Honorary Chair, General Chair, Program Chair, TPC Chair, Symposium Chair and Track Chair for many conferences, and was invited to give keynotes at numerous conferences.

Md Palash Uddin (Member, IEEE) earned a Ph.D. degree in IT from Deakin University, Australia. Dr. Uddin is currently a Postdoctoral Research Fellow at the School of IT, Deakin University, Australia, and also a faculty member at Hajee Mohammad Danesh Science and Technology University, Bangladesh. He has authored over 100 journal and conference papers in leading venues, including IEEE TIFS, IEEE TPDS, IEEE TSC, IEEE TMC, IEEE TCC, IEEE TNSE, IEEE TII, IEEE TBD, IEEE JSAC, IEEE SPL, and ACM CSUR. Additionally, he actively reviews for top-tier journals and conferences. His primary research interests lie in federated learning and classical machine learning, focusing on their applications in data privacy, security, and integrity verification.



Yushu Zhang (Senior Member, IEEE) received the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in December 2014. He held various research positions with The City University of Hong Kong, Hong Kong; Southwest University, Chongqing; the University of Macau, Macau, China; Deakin University, Geelong, VIC, Australia; and Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is currently a Professor with the School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics, Nanchang, China. His research interests include multimedia security, artificial intelligence security, and blockchain. He has published more than 150 refereed journal articles and conference papers in these areas. He is an Editor of Information Sciences, Journal of King Saud University-Computer and Information Sciences, and Signal Processing.

Dezhong Peng (Senior Member, IEEE) received the B.Sc. degree in applied mathematics and the M.Sc. and Ph.D. degrees in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 1998, 2001, and 2006, respectively. From 2001 to 2007, he was with the University of Electronic Science and Technology of China as an Assistant Lecturer and a Lecturer. He was a Post-Doctoral Research Fellow with the School of Engineering, Deakin University, Geelong, VIC, Australia, from 2007 to 2009. He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu, China. His research interests include neural networks and signal processing.

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cloud 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/TCC.2026.3697006

Jonathan Kua is currently a Senior Lecturer in Internet of Things at Deakin University. He received his B.Eng. (First Class Hons.) degree in telecommunications and network engineering in 2014 and Ph.D. degree in telecommunications engineering in 2019, both from Swinburne University of Technology, Melbourne, Australia. He was previously affiliated with the Centre for Advanced Internet Architectures (2013-2017) and the Internet For Things Research Group (2017-2019), within Swinburne University of Technology. He was awarded a full Netflix Ph.D. research scholarship to pursue applied research in high-performance Internet Protocol-based content delivery and adaptive streaming. His research was internationally recognized with a second runner-up of the DASH-IF Best Ph.D. Dissertation Award on “Algorithms and Protocols for Adaptive Content Delivery over the Internet” (2019), and commended locally with the Swinburne Outstanding Thesis Award (2019).

Abdorasoul Ghasemi is an Associate Professor of computer science at the Centre for Computational Science and Mathematical Modelling, Coventry University, UK. He received the B.Sc. (with highest honours) degree from the Isfahan University of Technology and the M.Sc. and PhD from the Amirkabir University of Technology (Tehran Polytechnique), Tehran, Iran, all in electrical and computer engineering. He has spent sabbatical leaves with Computer Science Department, University of California, Davis, CA, USA, from April 2017 to August 2018 and Max Planck Institute, Dresden, Germany, for the physics of complex systems from December 2020 to July 2021. He received the Alexander von Humboldt Fellowship for experienced researchers working on resilient cyber-physical energy systems with the University of Passau, Passau, Germany, from July 2021 to Dec. 2023. His research interests include network science and its engineering applications, including communications, energy, and cyber-physical systems using optimization and machine learning approaches. His research focuses on computation, communication, and control (C3) for networked infrastructure systems, particularly energy and communication systems.

Authorized licensed use limited to: Deakin University. Downloaded on June 23,2026 at 12:27:28 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.