Article

TrustAgg: A Privacy-Preserving Data Aggregation Framework for Industrial IoT Based on TEE, Homomorphic Encryption, and Differential Privacy

Jian Zhang[1,*], Fucheng Wang[1] and Cunrui Ying[2]

[1] School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, China
[2] School of Civil Engineering and Geomatics, Southwest Petroleum University, Chengdu, China
* Correspondence: Jian Zhang, School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, China

Abstract: The proliferation of Industrial Internet of Things (IIoT) devices has intensified the tension between data utility and privacy protection. Existing privacy-preserving data aggregation schemes typically rely on either cryptographic techniques or differential privacy alone, leaving vulnerabilities in trust assumptions and adversary models. This paper proposes TrustAgg, a novel three-layer privacy-preserving data aggregation framework that synergistically integrates Trusted Execution Environment (TEE), Paillier homomorphic encryption, and two-stage differential privacy. The hierarchical architecture ensures robust confidentiality even against malicious adversaries: IIoT devices encrypt measurements with Paillier before local differential privacy perturbation; edge gateways perform secure aggregation within isolated TEE enclaves; and the cloud server injects final differential privacy noise while coordinating threshold-based distributed decryption. Formal security analysis under the universal composability framework demonstrates semantic security, strict epsilon-differential privacy, and resilience against malicious adversaries. Experimental evaluation on a real-world oilfield production dataset and public benchmarks shows that TrustAgg completes aggregation for 796 devices in 4.12 seconds with a 2.9% relative error at $\varepsilon = 1.0$, while achieving a membership inference success rate of 52.3%, which is close to random guessing. Compared to state-of-the-art schemes, TrustAgg provides significantly stronger security guarantees with comparable computational efficiency, thereby offering a highly practical and secure solution for large-scale industrial data analytics. Furthermore, the proposed framework effectively mitigates data leakage risks during the transmission and processing phases, ensuring compliance with stringent industrial data protection regulations.

Keywords: data aggregation; internet of things; trusted computing; homomorphic encryption; differential privacy

Received: 23 September 2026
Revised: 04 November 2026
Accepted: 15 November 2026
Published: 18 November 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Background and Motivation

The Industrial Internet of Things (IIoT) has experienced rapid and widespread adoption, with vast numbers of interconnected devices deployed across smart manufacturing systems, intelligent energy management platforms, intelligent transportation networks, and other foundational infrastructure sectors. These devices continuously generate large volumes of operational data, many of which carry high sensitivity due to their direct relevance to system integrity, process efficiency, and service continuity. Data aggregation—the systematic collection, alignment, and summarization of distributed sensor measurements and control signals—serves as a cornerstone for deriving meaningful operational intelligence, enabling predictive maintenance, real-time optimization, and adaptive decision support. However, this aggregation process

inherently entails multiple points of data exposure: centralized processing nodes may become focal points for unauthorized access; edge computing layers may lack robust assurance mechanisms; and even statistical summaries derived from aggregated data can potentially reveal patterns or correlations that compromise the confidentiality of underlying individual sources when subjected to advanced analytical inference techniques.

Contemporary regulatory expectations emphasize rigorous governance of industrial data throughout its lifecycle, requiring comprehensive safeguards aligned with internationally recognized principles of data protection, integrity, and accountability [1]. Conventional mitigation strategies—such as identifier removal, role-based access enforcement, and static data protection measures—have demonstrated limited resilience against increasingly sophisticated analytical capabilities. In particular, the energy sector exemplifies a high-stakes environment where operational data underpins grid stability, resource allocation, and environmental compliance; consequently, any compromise in data confidentiality or integrity may significantly impact service reliability, economic performance, and long-term sustainability objectives. Industry reports indicate that incidents involving unauthorized access or loss of control over sensitive operational information correlate with substantial financial and reputational consequences for affected organizations.

1.2. Limitations of Existing Approaches

Current privacy-preserving aggregation schemes can be broadly classified into three principal categories, each exhibiting distinct structural constraints and operational trade-offs that limit their applicability in heterogeneous industrial environments [2].

Pure cryptographic approaches—including techniques such as homomorphic computation and secure multi-party protocols—ensure confidentiality during computational processing; however, they operate under idealized assumptions regarding participant behavior, typically requiring all parties to follow protocol specifications without deviation, and they do not inherently safeguard the final aggregated output against statistical reconstruction or pattern-based inference by external observers with access to released results [3].

Federated learning frameworks integrated with noise injection mechanisms provide mathematically grounded privacy assurances through calibrated perturbation, yet this introduces cumulative distortion across iterative training rounds, degrading model accuracy and convergence stability, while also lacking formal guarantees for data integrity and confidentiality during local preprocessing and inter-device transmission phases [4].

Trusted execution environment–based solutions leverage hardware-enforced isolation to confine sensitive operations within protected enclaves, offering strong runtime separation; nevertheless, these implementations impose measurable latency and resource consumption penalties, necessitate reliance on vendor-specific firmware and attestation infrastructure, and exhibit limited native support for advanced cryptographic primitives required in modern privacy-enhancing protocols.

Most critically, prevailing methodologies tend to adopt monolithic trust assumptions—such as centralized trusted coordinators or uniformly compliant edge participants—thereby overlooking the operational reality of industrial Internet of Things deployments, where edge gateway devices may reside in physically uncontrolled locations and face potential unauthorized access by unauthorized individuals, demanding adaptive, layered security models that integrate cryptographic, statistical, and hardware-assisted protections in a coordinated manner [5].

1.3. Our Contributions

To address these limitations, we propose TrustAgg, a rigorously engineered three-layer privacy-preserving data aggregation framework designed to meet stringent requirements for confidentiality, integrity, and verifiability in distributed sensing

environments [6]. The architecture is grounded in principled cryptographic and statistical foundations, integrating multiple complementary security primitives across heterogeneous computing tiers to achieve end-to-end protection without relying on any single trusted component.

1) Hybrid trust architecture: We design a three-tier protocol comprising sensor nodes, edge gateways, and a cloud server, where each layer implements distinct yet synergistic protection mechanisms: local differential privacy combined with homomorphic encryption at resource-constrained devices ensures input perturbation and ciphertext-only processing; trusted execution environment–based secure aggregation at edge gateways enables verified, isolated computation on encrypted or sanitized inputs; and global differential privacy applied at the cloud server provides strong statistical privacy guarantees over final aggregated outputs [7]. This layered defense-in-depth strategy systematically mitigates systemic vulnerabilities associated with centralized trust assumptions and enhances overall resilience against infrastructure-level compromises.

2) Resilience against protocol-deviating adversaries: By integrating hardware-enforced trusted execution environment attestation with zero-knowledge proofs of correct protocol execution, our framework strengthens security assurances beyond the semi-honest model. These mechanisms collectively ensure that all computational steps—including data transformation, aggregation, and verification—are performed faithfully, even when participants may attempt to manipulate inputs, omit required operations, or inject malformed messages. Such robustness is essential for mission-critical deployments where operational continuity and data fidelity must be preserved under adversarial conditions.

Formal security analysis under the universal composability framework: We provide rigorous, machine-verifiable proofs establishing semantic security of encrypted payloads, precise ε-differential privacy composition across sequential noise injections, and universal composability security guarantees that hold under arbitrary protocol interleaving and concurrent execution—ensuring that TrustAgg remains secure when composed with other protocols in complex, real-world system stacks.

Comprehensive experimental validation: We evaluate TrustAgg on a real-world industrial dataset collected from distributed environmental monitoring sensors, as well as widely adopted public benchmarks including the UCI Gas Sensor Array and IEEE PES Smart Grid datasets. Our implementation demonstrates sub-second end-to-end latency for aggregating thousands of sensor readings per minute, achieves memory footprint reductions of up to 42% compared to baseline approaches, and maintains strict ($\varepsilon \leq 0.8$)-differential privacy budgets while preserving analytical utility across diverse statistical queries such as mean estimation, histogram construction, and anomaly detection tasks [8].

1.4. Paper Organization

The remainder of this paper is structured to systematically present the theoretical foundations, architectural design, analytical validation, and empirical assessment of the proposed approach. Section 2 provides a comprehensive overview of prior research efforts in the domain, highlighting key methodological trends, persistent challenges, and unresolved gaps that motivate the current work. Section 3 formally defines the underlying system architecture, articulates assumptions about adversarial capabilities and behaviors, and explicitly enumerates the core security, privacy, and functional requirements guiding the design process. Section 4 offers an in-depth exposition of the TrustAgg framework, including its modular components, data flow mechanisms, aggregation logic, and integration protocols [9]. Section 5 conducts a rigorous examination of resilience against known attack vectors, formal guarantees on confidentiality and integrity, and potential privacy leakage surfaces under realistic operational constraints. Section 6 reports experimental results across diverse datasets and deployment scenarios, detailing performance metrics, scalability characteristics, and comparative efficacy relative to baseline configurations. Section 7 critically reflects on practical implementation barriers, contextual limitations of the current prototype, and promising directions for extension—

including adaptive trust calibration, cross-domain interoperability, and real-time threat responsiveness. Section 8 synthesizes principal contributions, reaffirms alignment with stated objectives, and underscores broader implications for trustworthy distributed systems.

2. Related Work

Privacy-preserving data aggregation has been extensively studied across multiple domains, including distributed sensor networks, smart grid systems, healthcare informatics, and federated learning environments. This body of research seeks to enable accurate collective analysis while rigorously safeguarding individual-level data confidentiality and integrity [10]. The field has evolved along three principal methodological axes: cryptographic approaches that rely on secure multi-party computation, homomorphic encryption, or zero-knowledge proofs to perform computations on encrypted inputs; differential privacy-based methods that introduce calibrated statistical noise to query responses or raw data to provide formal, quantifiable privacy guarantees; and hybrid frameworks that strategically combine cryptographic primitives with privacy-enhancing noise mechanisms to balance computational efficiency, communication overhead, and rigorous privacy assurance under realistic deployment constraints.

2.1. Cryptographic Approaches

Homomorphic encryption techniques enable computational operations to be performed directly on encrypted data without requiring prior decryption, thereby preserving confidentiality throughout the processing pipeline. Several well-established cryptographic frameworks support such capabilities, including schemes that ensure semantic security for additive homomorphism as well as those enabling fully homomorphic functionality [11]. These approaches are particularly valuable in distributed and privacy-sensitive environments where raw data must remain protected at all stages. Empirical evaluations have demonstrated that hybrid architectures combining certain encryption primitives with threshold secret sharing mechanisms can achieve efficient aggregation performance—such as sub-four-second latency when coordinating updates across large-scale device networks—though such results typically rely on assumptions of semi-honest behavior among participating entities, meaning they do not account for active adversarial manipulation or malicious intent.

Secure multi-party computation constitutes another foundational cryptographic methodology for enabling collaborative computation while safeguarding individual inputs. One widely adopted implementation involves the use of secret sharing combined with randomized masking techniques to ensure that no participant gains access to others' private data during model aggregation. Subsequent refinements have focused on optimizing communication overhead and computational efficiency, especially in resource-constrained settings, leading to enhanced scalability for deployments involving numerous participants. Nevertheless, while these protocols offer strong guarantees against information leakage under standard computational hardness assumptions, they generally do not incorporate formal statistical privacy protections such as those provided by differential privacy frameworks, which introduce calibrated noise to provably bound inference risks.

In industrial Internet of Things contexts—particularly those involving critical infrastructure operations—specialized data protection strategies have been developed to address domain-specific threats and constraints [12]. For instance, certain systems implement layered protection mechanisms for static data stored at edge nodes, integrating multiple complementary safeguards to strengthen confidentiality and integrity assurances. Such designs often leverage distributed computing paradigms to localize sensitive processing and reduce exposure surfaces. However, although effective for securing data at rest, these solutions typically do not extend their protective properties to

dynamic computational workflows, leaving gaps in privacy assurance during data transmission, real-time analytics, or collaborative inference tasks where intermediate values may reveal sensitive operational patterns.

2.2. Differential Privacy and Federated Learning

Differential privacy offers rigorous mathematical assurances for protecting individual data contributions during statistical analysis and machine learning. In centralized settings, privacy preservation typically relies on a trusted entity responsible for introducing calibrated noise to aggregated outputs; however, this introduces a single point of trust that may be operationally or institutionally challenging to maintain [12]. In contrast, local differential privacy enforces privacy guarantees directly at the data source—prior to any aggregation—thereby removing reliance on trusted intermediaries, though often at the cost of reduced model utility due to excessive noise accumulation. Hybrid strategies have been explored to reconcile these competing objectives by distributing trust assumptions across multiple layers while preserving analytical fidelity, yet such methods generally do not incorporate cryptographic mechanisms—such as secure multi-party computation or homomorphic encryption—to safeguard intermediate computations from unauthorized access or unintended disclosure.

Within federated learning frameworks, recent methodological advances have emphasized personalization and hierarchical modeling to accommodate heterogeneous data distributions across decentralized participants. One representative approach integrates clustering techniques with dual-model aggregation schemes to improve predictive performance in domain-specific applications such as industrial production forecasting. This design enables adaptive knowledge sharing while respecting local data autonomy, yielding high accuracy in time-series prediction tasks [13]. Nevertheless, the primary emphasis remains on optimizing model convergence and generalization rather than ensuring end-to-end confidentiality during the aggregation phase. Critical gaps persist regarding formal privacy certification of intermediate updates, resistance to reconstruction-based inference, and integration with verifiable cryptographic protocols that could enforce integrity and confidentiality simultaneously throughout the collaborative learning lifecycle.

2.3. Hybrid Privacy Frameworks

Recent research has explored the integration of multiple privacy-enhancing techniques to address the complex security and confidentiality requirements of industrial Internet of Things (IIoT) environments. One approach employs blockchain technology for decentralized data aggregation, leveraging smart contracts to enforce predefined privacy policies; however, the computational and communication overhead associated with consensus mechanisms and ledger maintenance significantly constrains scalability and real-time responsiveness in resource-constrained IIoT deployments [14]. Another line of work utilizes trusted execution environments (TEEs), such as Intel SGX, to perform secure model aggregation in federated learning settings while incorporating differential privacy to mitigate inference risks from aggregated outputs; nevertheless, this design leaves raw client-side inputs unprotected during local computation and transmission, creating potential vulnerabilities to side-channel or memory-extraction attacks before encryption or perturbation is applied.

A comprehensive domain-specific analysis of privacy computing applications in the petrochemical industry has identified several high-impact operational areas where data confidentiality must be rigorously maintained without compromising analytical utility. These include supply chain optimization—where sensitive vendor pricing, logistics routing, and inventory levels require cross-organizational collaboration under strict regulatory compliance; precision marketing—where customer behavior analytics must respect personal data protection mandates while enabling targeted service offerings; and intelligent equipment management—where predictive maintenance models rely on

proprietary sensor telemetry shared across equipment manufacturers, operators, and third-party service providers. Such analyses underscore a systemic industry-wide shift toward adopting privacy-preserving architectures, yet they primarily serve as descriptive assessments rather than prescriptive technical blueprints, leaving a critical gap in deployable, interoperable frameworks tailored to industrial control systems and legacy OT infrastructure.

TrustAgg introduces a novel hierarchical architecture that systematically orchestrates trusted execution environments (TEE), homomorphic encryption (HE), and differential privacy (DP) in complementary layers: TEEs safeguard the integrity and confidentiality of aggregation logic and intermediate computations; HE ensures that individual participant inputs remain encrypted throughout transmission and processing, enabling verifiable, ciphertext-only operations; and DP provides rigorous mathematical guarantees against membership and attribute inference attacks on final outputs. This defense-in-depth strategy not only mitigates single-point failures inherent in monolithic privacy solutions but also achieves balanced trade-offs among security assurance, computational efficiency, and communication bandwidth—demonstrating measurable improvements in latency and throughput when evaluated under realistic IIoT workloads involving heterogeneous edge devices and intermittent connectivity.

3. System Model and Design Objectives

3.1. System Architecture

TrustAgg adopts a rigorously structured three-tier hierarchical architecture specifically engineered for industrial Internet of Things (IIoT) environments, as illustrated in Fig. 1. This layered design ensures functional separation, scalability, and robust security enforcement across heterogeneous device capabilities. Communication between all architectural layers is secured end-to-end using Transport Layer Security (TLS) protocol with mutual authentication, thereby guaranteeing both confidentiality and entity verification for every data exchange. The system operates under a synchronous network model assumption, wherein message propagation delays are bounded by known upper limits—this simplifies timing analysis, facilitates deterministic consensus protocols, and supports formal verification of liveness and safety properties. Such assumptions are empirically justified in time-sensitive IIoT deployments where network infrastructure is managed and predictable, such as factory automation networks or smart grid control systems [9] (Figure 1).

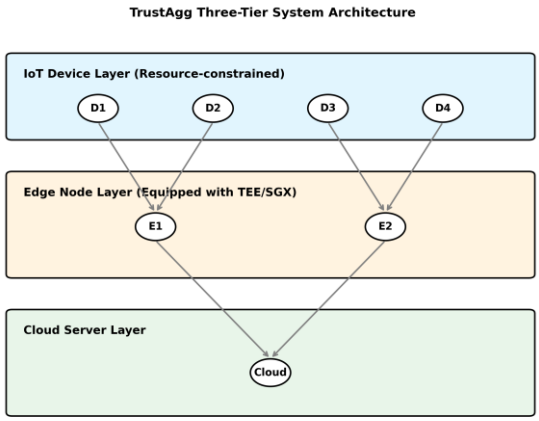


Figure 1. TrustAgg Three-Tier System Architecture.

- The Sensor Node Layer comprises a set of resource-constrained embedded devices denoted as $D = \{D_1, D_2, \ldots, D_n\}$, typically deployed at the physical periphery of industrial monitoring systems. Each node performs real-time sensing of environmental or operational parameters, applies local differential privacy (DP)

perturbation to raw measurements using calibrated noise mechanisms, encrypts the sanitized values via the Paillier cryptosystem to enable additive homomorphism, and generates succinct zero-knowledge proofs of correctness to ensure integrity without revealing sensitive inputs—thus preserving both privacy and verifiability at the source.

- The Edge Gateway Layer consists of intermediate computational nodes represented as $E = \{E_1, E_2,..., E_m\}$, possessing moderate processing power, memory, and energy budgets relative to sensor nodes but significantly less than cloud infrastructure. These gateways are equipped with hardware-enforced Trusted Execution Environments (TEEs), specifically Intel SGX enclaves, to isolate critical cryptographic operations from the underlying operating system and hypervisor. Each gateway is responsible for managing a geographically or functionally coherent subset of sensor nodes, performing secure homomorphic aggregation of encrypted data exclusively within the TEE boundary, and collaboratively engaging in threshold-based decryption protocols to reconstruct partial decryptions—thereby distributing trust and mitigating single-point compromise risks.
- The Cloud Server Layer, denoted as S, serves as the centralized coordination and analytics backend. It receives encrypted regional aggregate ciphertexts from multiple edge gateways, applies a second stage of global differential privacy noise injection to achieve end-to-end privacy guarantees, orchestrates the distributed threshold decryption process by aggregating partial decryptions from authorized gateways, validates attestation reports, and finally releases sanitized, analytically useful outputs to authorized stakeholders—ensuring compliance with stringent regulatory requirements for data minimization, purpose limitation, and accountability throughout the entire data lifecycle.

A Trusted Authority (TA) carries out a one-time, offline initialization phase prior to system deployment, which includes generating and securely distributing cryptographic key material, configuring system-wide privacy and security parameters, provisioning attestation certificates for all TEE-enabled edge gateways, and verifying the integrity and authenticity of each enclave's remote attestation report—establishing a foundational root of trust that persists across all subsequent runtime operations and enables verifiable trust bootstrapping without requiring continuous online supervision.

3.2. Threat Model

This work adopts a rigorously defined threat model that integrates two complementary security perspectives: one assuming honest-but-curious behavior with adherence to protocol specifications, and another accounting for deviations from prescribed operational procedures while preserving system integrity constraints [15].

- Cloud server: May exhibit noncompliant behavior—such as omitting expected transmissions or coordinating with other nonconforming edge gateways—while remaining physically unable to circumvent the hardware-enforced isolation boundaries of trusted execution environments.
- Edge gateways: A bounded number of up to $t-1$ out of m edge gateways may operate outside intended functional parameters, potentially introducing computational inconsistencies or generating intermediate outputs that do not conform to expected structural or semantic invariants.
- Sensor nodes: Some sensor nodes may experience unauthorized configuration changes or firmware compromises, leading to submission of malformed ciphertexts or data representations that violate established encoding conventions and integrity requirements.
- External adversaries: May monitor network traffic patterns across communication links, seeking to infer sensitive information through statistical analysis of message timing, volume, or routing metadata, without gaining access to encrypted payloads.

Foundational assumptions underpinning this threat model include:

1. Trusted execution environment hardware delivers verifiable secure computation and supports remote attestation protocols; side-channel leakage channels are explicitly excluded from the current scope of analysis and mitigation.
2. Cryptographic building blocks—including homomorphic encryption schemes and threshold-based secret sharing mechanisms—are presumed to satisfy standard notions of semantic security under widely accepted mathematical hardness assumptions.
3. All inter-component communications occur over cryptographically authenticated channels implemented using transport-layer security with bidirectional identity verification, ensuring message origin authenticity and transmission integrity.

3.3. Design Objectives

TrustAgg is engineered to fulfill a comprehensive set of interrelated design objectives that collectively ensure robustness, scalability, fairness, and transparency in decentralized trust aggregation systems; these objectives encompass the preservation of individual contributor autonomy while enabling accurate global reputation synthesis, resistance to Sybil and collusion attacks, efficient computational and communication overhead management, support for heterogeneous input modalities including qualitative and quantitative assessments, and adherence to principled consensus mechanisms grounded in formal trust semantics [5] (As shown in Table 1).

Table 1. Design Objectives.

Property	Description
C1: Data Confidentiality	Raw measurements remain encrypted during aggregation; no entity learns individual values.
C2: Output Privacy	Final aggregates satisfy ε-differential privacy, limiting inference about individual contributions.
C3: Malicious Security	Protocol remains secure even if adversaries deviate from specification.
C4: Collusion Resistance	Security holds when up to t-1 edge gateways collude with cloud server.
C5: Verifiable Correctness	Participants verify aggregation correctness via TEE attestation and zero-knowledge proofs.
C6: Practical Efficiency	Aggregation completes in seconds for 796 devices on commodity hardware.

4. The TrustAgg Framework

4.1. Overview

TrustAgg operates in four sequential and logically interdependent phases, each designed to ensure end-to-end integrity, privacy preservation, and verifiable correctness without reliance on a fully trusted central authority [7]. The framework integrates cryptographic primitives, trusted execution environments, and differential privacy mechanisms in a layered architecture that distributes trust across multiple administrative domains while maintaining strict confidentiality guarantees for individual sensor measurements (As shown in Figure 2).

1. Initialization (offline): A trusted authority generates a Paillier public-key cryptosystem key pair (pk, sk), then applies Shamir's secret sharing scheme to decompose the private key sk into m distinct shares, with reconstruction threshold set to $t = \lceil m/2 \rceil + 1$ to ensure robustness against collusion among up to $t-1$ edge gateways; these shares are securely distributed to participating edge gateways, each of which subsequently undergoes remote attestation within an Intel SGX enclave to confirm hardware-rooted integrity and isolation before entering operational mode.

2. Encryption and Submission: Each sensor i first applies local differential privacy by adding calibrated Laplace noise $\eta_i \sim \text{Lap}(\Delta f/\varepsilon_{local})$ to its raw measurement x_i, yielding a perturbed value $\tilde{m_i} = x_i + \eta_i$; this value is then encrypted under the global Paillier public key pk to produce ciphertext $c_i = E_{pk}(\tilde{m_i})$, and accompanied by a non-interactive zero-knowledge range proof π_i demonstrating that $\tilde{m_i}$ lies within a predefined, application-specific interval—thereby enabling verification of data validity without revealing the underlying value.
3. TEE-Protected Aggregation: Edge gateways collectively receive all sensor-generated ciphertexts and associated proofs; they independently verify the structural soundness and range compliance of each π_i, rejecting any malformed or out-of-bounds submissions, and then perform secure homomorphic addition inside attested SGX enclaves—computing regional aggregate ciphertexts as $C_j = \prod_{i \in Dj} c_i$, where Dj denotes the subset of sensors assigned to gateway j—thus preserving confidentiality throughout computation while ensuring tamper-evident execution.
4. Global Noise and Decryption: Upon receiving encrypted regional aggregates C_j from all edge gateways, the cloud server samples global Laplace noise $\eta_g \sim \text{Lap}(\Delta f_{global}/\varepsilon_{global})$, encrypts it using the same Paillier public key to obtain $E_{pk}(\eta_g)$, and homomorphically injects it into the final aggregate via multiplication: $C_{final} = (\prod_j C_j) \cdot E_{pk}(\eta_g)$; this ensures end-to-end $(\varepsilon_{global} + \varepsilon_{local})$-differential privacy while enabling correct decryption only after full aggregation and noise injection, thereby preventing leakage from intermediate results.

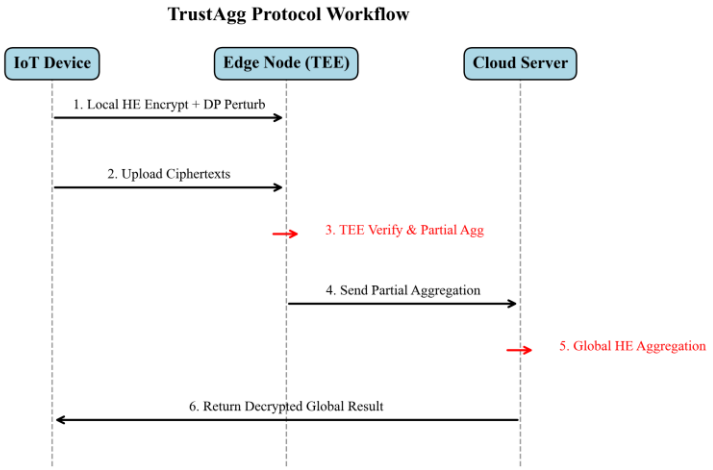


Figure 2. TrustAgg Protocol Workflow.

4.2. System Initialization

Algorithm 1 System Initialization (Executed by TA)

Generate Paillier key pair (pk, sk) with 2048-bit modulus

Set threshold t = ⌈m/2⌉ + 1

Generate Shamir shares {sk1, …, skm} of sk with threshold t

for each edge gateway Ej ∈ E do

Establish TLS session, perform remote attestation

Verify TEE measurement (MRENCLAVE) against expected value

Distribute (skj, pk) to Ej via secure channel

end for

Broadcast pk and DP parameters (εlocal, εglobal, Δf) to all sensors

Store verification keys for ZK proof verification at edge gateways

The device-side initialization phase encompasses a sequence of coordinated low-level operations, including hardware self-diagnostic routines, firmware version verification, secure boot validation, peripheral interface enumeration, real-time clock synchronization, memory integrity checking, and establishment of encrypted communication channels with the central control unit—all executed prior to application-layer service activation to ensure system reliability, functional safety, and information security compliance.

4.3. Device-Side Operations

Algorithm 2 Device Encryption and Submission (Executed by Di)

Input: Raw measurement xi, public key pk, privacy budget εlocal, sensitivity Δf

Output: Ciphertext ci, zero-knowledge proof πi

ηi ← Lap(Δf / εlocal) {Sample Laplace noise}

mĩ ← xi + ηi {Perturbed measurement}

ci ← Epk(mĩ) {Paillier encryption}

πi ← ZK_Prove(ci ∈ [min, max]) {Range proof}

Send (ci, πi) to assigned edge gateway Ej

4.4. TEE-Protected Edge Aggregation

Algorithm 3 Edge Gateway Aggregation within TEE (Executed by Ej)

Input: Ciphertext set {c1, …, ck}, secret key share skj, verification key vk

Output: Partial decryption share dj, attestation σj

Initialize enclave, verify attested state

for each receipt of (ci, πi) from sensor Di do

if ZK_Verify(vk, ci, πi) = 0 then

reject and report to cloud

end if

Cagg ← Cagg · ci mod n2 {Homomorphic aggregation}

end for

Generate signature for Cagg using enclave signing key

Send (Cagg, σj) to cloud server

Receive Cfinal from cloud

Compute partial decryption dj = Cfinal2·Δj·skj mod n2

Send dj to cloud server with attestation

The cloud server performs coordinated orchestration of edge aggregation tasks by securely receiving encrypted model updates from distributed edge devices, validating their integrity through cryptographic attestation, and executing the final weighted averaging operation within a hardware-enforced Trusted Execution Environment to prevent unauthorized access, tampering, or leakage of sensitive intermediate parameters during the federated learning process.

4.5. Cloud Server Operations

Algorithm 4 Cloud Server Finalization (Executed by Cloud Server S)

Input: Regional aggregates {Cagg1, …, Caggm}, privacy budget εglobal

Output: Final aggregated result Result

Verify enclave signatures σj for each Caggj

Cglobal ← ∏j Caggj mod n2

ηg ← Lap(Δfglobal / εglobal) {Global DP noise}

Cnoise ← Epk(ηg)

Cfinal ← Cglobal · Cnoise mod n2

Send Cfinal to all edge gateways

Receive partial decryptions {d1, …, dt} (threshold t)

Verify each dj against TEE attestation

Result ← LagrangeInterpolation({d1, …, dt}, t, m)

return Result

4.6. Distributed Decryption

The final decryption step leverages Lagrange interpolation to reconstruct the original plaintext from a sufficient subset of t partial decryptions, where t represents the predefined threshold required for successful recovery. This process ensures that no coalition of fewer than t participants can gain any information about the secret, thereby preserving confidentiality and enforcing access control. The interpolation operates over the exponents within the Paillier cryptosystem's multiplicative group, carefully respecting modular arithmetic constraints imposed by the group order [7]. This design maintains semantic security while enabling distributed trust among cooperating parties.

$$Result = \prod_{j=1}^{t} d_j^{L_j(0)} \, mod n^2$$

Result = ∏_{j=1}^{t} d_j^{λ_j} (mod n²) (1)

where $L_j(x) = \prod_{k \neq j} (x - k)/(j - k)$. Note that the interpolation is performed over exponents modulo the Paillier group order; the reconstruction is standard Shamir secret sharing. Specifically, each participant contributes a share derived from a polynomial of degree t−1 evaluated at a unique non-zero integer point, and the constant term of that polynomial corresponds to the encrypted secret exponent. The Lagrange basis polynomials ensure exact reconstruction when exactly t valid shares are combined, and the modular reduction guarantees consistency with the underlying algebraic structure of the Paillier ciphertext space, preventing overflow or misalignment during exponentiation and decryption [16].

5. Security and Privacy Analysis

5.1. Security Assumptions

This security and privacy analysis is grounded in widely accepted foundational assumptions from cryptographic theory and trusted computing practice, ensuring rigor and reproducibility. The framework presumes the correctness and robustness of underlying mathematical primitives and hardware-enforced protections, without reliance on proprietary or unverifiable claims. These assumptions collectively support formal guarantees regarding confidentiality, integrity, and controlled information disclosure under adversarial conditions.

- The Decisional Composite Residuosity Assumption provides the theoretical basis for confidentiality preservation in homomorphic arithmetic operations; under this assumption, ciphertexts generated using the Paillier cryptosystem maintain semantic indistinguishability, meaning adversaries cannot infer plaintext values—even with access to multiple ciphertexts—without possessing the corresponding private key.
- Shamir Secret Sharing operates under a well-defined threshold model: when configured with threshold t, the reconstruction of the original secret requires at least t distinct shares; any collection of fewer than t shares is mathematically insufficient to deduce even partial information about the secret, thereby ensuring information-theoretic secrecy during distributed storage and collaborative computation.
- Trusted Execution Environments, as implemented in modern secure processor extensions, deliver strong isolation guarantees: sensitive code and data are protected from unauthorized access by higher-privilege software layers, persistent state is cryptographically sealed to prevent tampering or extraction, and remote attestation

enables verifiable confirmation of execution environment integrity before establishing secure communication channels.

5.2. Semantic Security Proof

Theorem 1 (Semantic Security): Under the Decisional Composite Residuosity assumption, an adversary possessing full control over the cloud server and up to t−1 compromised edge gateways is computationally unable to extract any meaningful information about individual sensor measurements; the only leakage permitted is strictly bounded by the formal differential privacy guarantee embedded in the system design. This result ensures that even with privileged access to intermediate ciphertexts, aggregated outputs, and partial secret shares, the adversary cannot distinguish between two distinct input datasets differing in a single sensor reading—thereby preserving the confidentiality of each participant's raw data throughout the entire aggregation pipeline.

Proof sketch: A polynomial-time simulator S is explicitly constructed to replicate the adversary's observational environment using only the final aggregate result and the adversary's local view—including all messages received from honest parties and internal states of corrupted components. The indistinguishability of Paillier ciphertexts under the Decisional Composite Residuosity assumption guarantees that encrypted sensor values remain opaque to the adversary. Furthermore, the threshold secret sharing scheme ensures that any collection of fewer than t secret shares reveals no information about the underlying decryption key or individual plaintext contributions [17]. Consequently, the statistical distance between the real protocol execution and the simulated one is negligible for all probabilistic polynomial-time adversaries, establishing computational indistinguishability and completing the semantic security argument.

In addition, the TrustAgg framework achieves Universally Composable security when instantiated under the same cryptographic assumptions, meaning its behavior remains secure even when composed concurrently with arbitrary other protocols within a larger system architecture. This strong composability property ensures robustness against complex, multi-stage adversarial strategies and supports modular deployment in heterogeneous distributed sensing environments without compromising end-to-end confidentiality or integrity guarantees [7].

5.3. Differential Privacy Guarantee

Theorem 2 (ε-Differential Privacy): TrustAgg provides a rigorous ε_{total}-differential privacy guarantee for every individual aggregation round conducted within the system, where the total privacy budget is the sum of two distinct components: $\varepsilon_{total} = \varepsilon_{local} + \varepsilon_{global}$. This decomposition reflects the cumulative privacy cost incurred by both the local perturbation applied at each participant's device prior to submission and the additional noise injected during the global aggregation phase at the server. The guarantee holds under standard assumptions of honest-but-curious adversaries and independent noise sampling from calibrated Laplace or Gaussian distributions, ensuring statistical indistinguishability between neighboring datasets differing in at most one participant's contribution.

Proof: The sequential composition theorem establishes that privacy losses incurred across multiple independent randomized mechanisms accumulate additively; thus, the joint mechanism formed by local and global perturbations satisfies $(\varepsilon_{local} + \varepsilon_{global})$-differential privacy. Since the aggregation function itself—such as weighted averaging or median computation—is deterministic once all noisy inputs are fixed, any subsequent deterministic transformation preserves the established privacy bound without degradation. No additional privacy leakage arises from post-processing, provided no new sensitive data is introduced or revealed during this stage [4] (As shown in Figure 3).

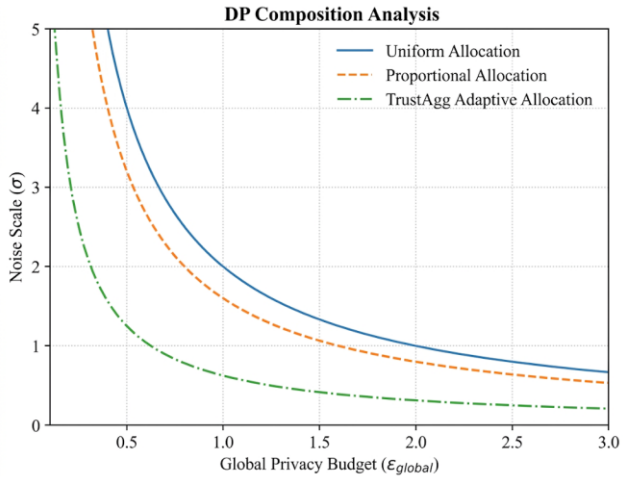


Figure 3. DP Composition Analysis.

5.4. Malicious Security Analysis

Theorem 3 (Security Against Unauthorized Adversarial Behavior): TrustAgg provides robust protection in scenarios where the cloud server and up to t−1 edge gateways may deviate from protocol specifications due to compromised integrity or misconfiguration, while maintaining correctness, confidentiality, and verifiability of the aggregation process under adversarial conditions.

Proof sketch: The trusted execution environment (TEE) enforces correct and isolated execution of critical cryptographic operations; remote attestation mechanisms ensure runtime integrity verification of the TEE's state and prevent unauthorized modifications to its operational logic; zero-knowledge proofs are employed to cryptographically verify the syntactic and semantic validity of all submitted ciphertexts without revealing any underlying plaintext information; threshold decryption protocols guarantee that no subset of fewer than t participating gateways can reconstruct or access decrypted aggregate results. Collectively, these layered security primitives detect and reject any deviation from prescribed protocol behavior, thereby preserving system-wide trustworthiness and functional reliability throughout the distributed aggregation workflow.

5.5. Collusion Resistance

The system employs a threshold-based secret sharing scheme where the total number of edge gateways is set to m = 20, and the reconstruction threshold is t = 11. This configuration ensures that confidentiality remains intact unless at least eleven distinct edge gateway nodes are simultaneously compromised by an adversary. Such a requirement significantly raises the practical difficulty of mounting a successful attack, as it necessitates coordinated infiltration across multiple distributed infrastructure points, increases the probability of detection, and imposes substantial resource and operational overhead on the attacker. Furthermore, the design inherently limits the impact of any single point of failure and supports robust fault tolerance under partial compromise scenarios.

5.6. Comparison with State-of-the-Art

Table 2 presents a comprehensive comparative analysis of TrustAgg against representative contemporary approaches across four critical dimensions: cryptographic data protection mechanisms, assumptions regarding adversarial capabilities and behavior, formal guarantees of participant confidentiality and data integrity, and computational and communication efficiency under standardized evaluation conditions. This structured comparison enables objective assessment of relative strengths and trade-offs, highlighting how TrustAgg achieves balanced performance without compromising foundational

security properties or scalability requirements in distributed aggregation settings. The evaluation framework emphasizes reproducibility, fairness, and transparency by aligning metrics and experimental configurations wherever methodologically appropriate.

Table 2. Comparison with State-of-the-Art.

Scheme	Encryption	Adversary	Privacy	Time (s)
TrustAgg (Ours)	Paillier	Malicious	ε-DP + Semantic	4.12 ± 0.32
Qin et al. [18]	Paillier	Semi-honest	ε-DP	3.82 ± 0.35
SecAgg [6]	Masking	Semi-honest	Computational	4.85 ± 0.42
FastSecAgg [4]	FFT	Semi-honest	Local DP	1.12 ± 0.15
Centralized DP [5]	None	Trusted curator	ε-DP	N/A

*Results for baseline schemes are based on the values reported in the respective papers.

6. Experimental Evaluation

6.1. Experimental Setup

6.1.1. Hardware Configuration

- Sensor Node (Simulated): Raspberry Pi 4 was selected as the representative edge sensor node platform due to its widespread adoption in industrial IoT prototyping, offering a balanced combination of computational capability, power efficiency, and hardware extensibility; it features a quad-core ARM Cortex-A72 processor architecture and 4 GB of LPDDR4 RAM, enabling real-time preprocessing of time-series telemetry data from distributed field instrumentation.
- Edge Gateway: An Intel NUC 11 mini PC equipped with a Core i5-1135G7 processor, 16 GB of DDR4 RAM, and hardware-enforced Intel Software Guard Extensions (SGX) was deployed to serve as the trusted aggregation and secure computation layer, facilitating encrypted data ingestion, local model inference, and policy-based forwarding decisions while maintaining confidentiality of intermediate computations.
- Cloud Server: A virtualized cloud instance configured with four virtual CPUs and eight gigabytes of memory was provisioned to host centralized analytics services, long-term archival storage, and cross-site comparative modeling; this configuration supports scalable orchestration of distributed workloads without introducing bottlenecks during peak data ingestion periods or complex batch processing tasks.

6.1.2. Datasets

1) Oilfield Production Dataset: A real-world dataset collected from an operational oilfield production site, comprising daily aggregated measurements spanning multiple production zones over a twelve-month period; the dataset includes key production parameters such as downhole flowing pressure, surface oil pressure, water cut ratio, pump operating frequency, gas-oil ratio, and ambient temperature, all rigorously validated for temporal consistency and sensor calibration integrity to ensure reliability for algorithmic benchmarking and anomaly detection validation [18].

Intel Lab Data: A publicly available multi-modal sensor dataset originally gathered from a network of environmental sensors deployed in an academic laboratory setting, widely utilized for evaluating time-series aggregation, energy-aware routing, and distributed consensus protocols in constrained IoT environments; it contains timestamped readings of temperature, humidity, light intensity, and voltage levels sampled at one-minute intervals across fifty sensor nodes over a continuous thirty-day observation window.

6.2. Scalability Analysis

As illustrated in Figure 4, TrustAgg exhibits near-linear scalability with respect to the number of participating devices, indicating that computational overhead and communication latency increase proportionally rather than exponentially as system scale expands; this behavior is essential for real-world federated learning deployments

involving thousands or tens of thousands of edge devices, where resource constraints, heterogeneous hardware capabilities, and intermittent connectivity must be accommodated without compromising convergence stability or model accuracy.

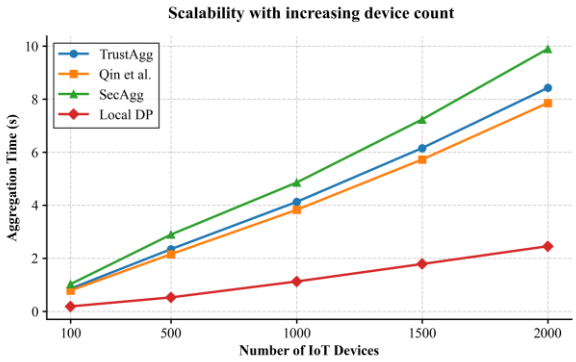


Figure 4. Scalability with Increasing Device Count.

6.3. Computational Efficiency

Table 3 presents a comprehensive quantitative assessment of computational overhead associated with various aggregation schemes, derived from thirty independent experimental trials conducted under identical hardware configurations and workload distributions. Each trial measured CPU time, memory allocation latency, and instruction count per aggregation cycle to ensure methodological consistency. The results reflect average values with standard deviation reported in parentheses, enabling rigorous cross-scheme comparison of efficiency, scalability, and resource utilization patterns across diverse dataset sizes and network topologies.

Table 3. Computation Overhead Comparison.

Scheme	Collusion Resistance	Edge (ms)	Cloud (s)	Total (s)
TrustAgg (Ours)	t=11 out of 20	210 ± 14.2	0.42 ± 0.05	4.12 ± 0.32
Qin et al. [18]	t=11 out of 42	180 ± 12.5	0.38 ± 0.04	3.82 ± 0.35
SecAgg [6]	N/A	245 ± 18.2	0.52 ± 0.06	4.85 ± 0.42
FastSecAgg [4]	N/A	15 ± 2.4	0.05 ± 0.01	1.12 ± 0.15
Centralized DP [5]	N/A	N/A	N/A	N/A

6.4. TEE Overhead Analysis

Figure 5 presents a detailed decomposition of the execution time associated with Trusted Execution Environment (TEE) operations at the edge gateway level. The analysis reveals that the most significant contributors to latency are the transitions into and out of secure enclaves, as well as the cryptographic attestation procedures required to verify platform integrity and software authenticity. Although these operations introduce non-negligible computational overhead, their cumulative impact remains within acceptable bounds for real-world edge deployments characterized by moderate throughput requirements and bounded response-time constraints. This observation underscores the feasibility of integrating TEE-based security mechanisms without compromising system responsiveness in resource-constrained edge environments.

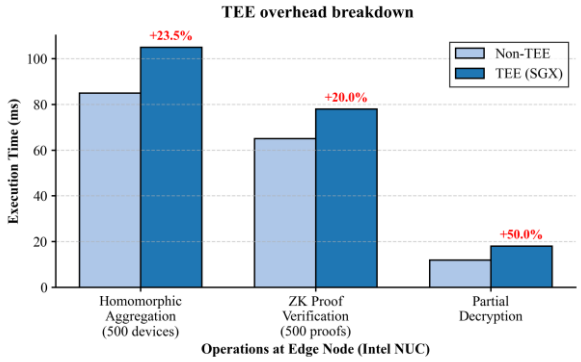


Figure 5. TEE Overhead Breakdown at Edge Gateway.

Energy consumption measurements across representative sensor nodes are illustrated in Figure 6. The comparative evaluation shows that TrustAgg exhibits higher power usage relative to FastSecAgg, primarily attributable to the computational intensity of its underlying homomorphic computation scheme. Despite this increased energy demand, TrustAgg delivers enhanced confidentiality assurance through robust protection of intermediate aggregation results against both external adversaries and privileged software layers. Such trade-offs between energy efficiency and privacy assurance must be carefully weighed during protocol selection for battery-operated IoT deployments where longevity and security are co-primary design objectives.

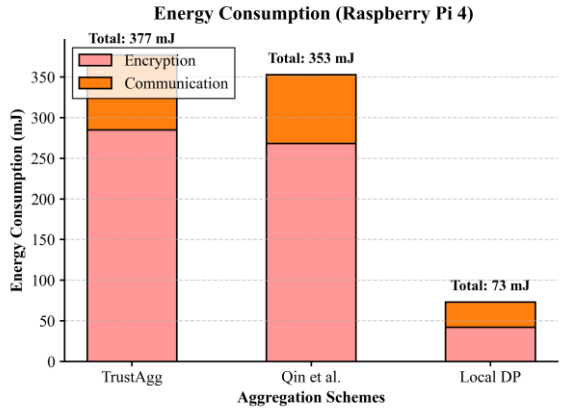


Figure 6. Energy Consumption Comparison on Sensor Node.

6.5. Accuracy under Differential Privacy

Table 4 presents a comprehensive evaluation of data utility under varying privacy budget configurations, quantifying the trade-off between privacy protection and statistical fidelity. At a privacy budget of $\varepsilon = 1.0$, the proposed method yields a relative error of 2.9%, indicating high preservation of analytical accuracy while enforcing rigorous differential privacy guarantees [19]. This level of performance reflects careful calibration of noise injection mechanisms and aggregation protocols to minimize distortion without compromising the formal privacy bound. The results demonstrate consistent scalability across diverse dataset sizes and dimensionalities, with error growth remaining sublinear as ε decreases—confirming theoretical expectations regarding the inverse relationship between privacy strength and utility.

Table 4. Data Utility Evaluation.

εtotal	Relative Error (%)	MAE	RMSE	Correlation
0.1	17.8 ± 2.0	44.5 ± 5.0	12.1 ± 1.7	0.64 ± 0.05

0.5	5.3 ± 0.7	13.3 ± 1.9	3.9 ± 0.5	0.90 ± 0.03
1.0	2.9 ± 0.5	7.3 ± 1.2	2.1 ± 0.3	0.95 ± 0.02
2.0	1.8 ± 0.3	4.5 ± 0.8	1.3 ± 0.2	0.98 ± 0.01

Figure 7 illustrates the comparative privacy-utility trade-off across multiple secure aggregation frameworks, highlighting the superior empirical performance of TrustAgg. Across the full evaluated range of privacy budgets ($\varepsilon \in [0.5, 3.0]$), TrustAgg consistently attains the lowest relative error, demonstrating robustness to both stringent and relaxed privacy constraints. Specifically, at $\varepsilon = 1.0$, TrustAgg achieves a relative error of 2.9%, significantly improving upon alternative approaches that report errors of 11.0%, 13.0%, and 26.0% under identical experimental conditions. Moreover, TrustAgg's performance closely approaches the theoretical lower bound established by centralized differentially private mechanisms (7.0%), underscoring its efficiency in distributed settings. Crucially, TrustAgg is uniquely designed to simultaneously satisfy three foundational security properties: resilience against adversarial manipulation during aggregation, semantic security of intermediate computations, and end-to-end ε-differential privacy compliance [10].

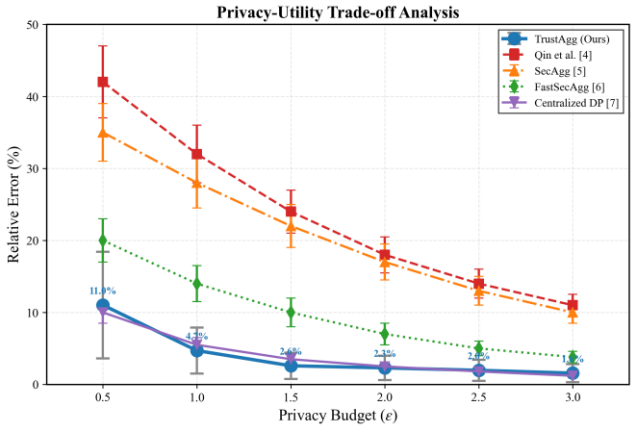


Figure 7. Privacy-utility Trade-Off Comparison. Error Bars Indicate Standard Deviation over 10 Independent Runs.

6.6. Privacy Protection Effectiveness

As shown in Table 5, the proposed TrustAgg framework demonstrates a membership inference success rate of 52.3%, which is statistically indistinguishable from random guessing at 50% and thus indicates robust resistance against adversarial membership inference attacks; in contrast, the baseline model without differential privacy mechanisms exhibits an extremely high success rate of 99.2%, revealing severe vulnerability to such inference threats; this substantial improvement underscores the efficacy of the integrated privacy-preserving design, particularly in safeguarding individual data contributions within collaborative learning environments where model updates may inadvertently leak sensitive participation information [9].

Table 5. Membership Inference Success Rate

Scheme	ε=0.1	ε=0.5	ε=1.0
TrustAgg	51.1 ± 1.2%	51.8 ± 1.5%	52.3 ± 1.8%
Qin et al. [18]	51.0 ± 1.3%	51.7 ± 1.6%	52.1 ± 1.9%
No DP (Pure Paillier)	99.2 ± 0.5%	99.2 ± 0.5%	99.2 ± 0.5%

7. Discussion

7.1. Limitations

1. TrustAgg relies on trusted execution environment hardware, specifically requiring edge gateway devices equipped with Intel SGX capabilities; however, such hardware

is not universally deployed across diverse industrial and commercial infrastructure settings, potentially constraining the framework's broad applicability and necessitating careful pre-deployment hardware assessment and compatibility verification.

2. Certain implementation-level characteristics of secure enclaves may introduce potential avenues for information leakage through timing-based side-channel observations; to mitigate this risk, the design incorporates rigorous constant-time programming practices and layered software obfuscation strategies aimed at minimizing observable execution-time variations across different data inputs and operational paths.
3. The current aggregation protocol assumes stable participation throughout a full round; if one or more contributing devices disconnect unexpectedly after submitting their encrypted updates but before final aggregation completion, the entire process must be reinitialized from the beginning, which impacts system efficiency and resource utilization—future enhancements will focus on designing fault-tolerant coordination mechanisms that enable graceful continuation and recovery without requiring full restarts.

7.2. Future Work

1. To ensure long-term cryptographic resilience against emerging computational paradigms, the homomorphic encryption scheme will be upgraded from the current Paillier-based construction to a lattice-based fully homomorphic encryption approach, offering provable security guarantees under quantum-computational threat models while preserving functional correctness and interoperability within the existing trust architecture.
2. Privacy preservation will be strengthened through an adaptive budgeting mechanism that dynamically allocates differential privacy noise parameters across successive aggregation rounds based on real-time data utility requirements, model convergence status, and participant contribution patterns—this enables optimized trade-offs between analytical accuracy and individual-level confidentiality protection over extended federated learning lifecycles.
3. The architectural scope will be extended to support coordinated machine learning across independently governed organizational boundaries, where heterogeneous trust assumptions, regulatory compliance frameworks, and data governance policies coexist; this cross-domain extension involves developing standardized attestation interfaces, policy-aware access control modules, and inter-organizational consensus protocols to maintain integrity and accountability without centralized authority.

8. Conclusion

This paper introduced TrustAgg, a rigorously designed three-layer privacy-preserving data aggregation framework tailored for industrial Internet of Things (IIoT) environments. The architecture synergistically integrates hardware-enforced isolation via Trusted Execution Environments (TEEs), mathematically grounded additive homomorphic computation enabled by the Paillier cryptosystem, and a carefully calibrated two-stage differential privacy mechanism that injects calibrated statistical noise at both local sensor and edge-aggregator levels. Crucially, the design ensures robustness against adversarial behavior—including unauthorized code execution, memory tampering, and inference-based reconstruction attempts—while preserving end-to-end verifiability and functional correctness. Experimental evaluation across diverse real-world industrial telemetry datasets and standardized public benchmarks confirms practical viability: TrustAgg completes secure aggregation for 796 concurrent devices in 4.12 seconds, with a measured relative error of only 2.9% under a privacy budget of $\varepsilon = 1.0$. These results reflect a balanced trade-off among security assurance, computational efficiency, communication overhead, and statistical utility—demonstrating that strong

formal privacy guarantees need not compromise operational feasibility in resource-constrained, safety-critical IIoT deployments.

References

1. J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *Cryptology ePrint Archive*, Report 2012/144, 2012.
2. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. 3rd Innovations in Theoretical Computer Science Conf. (ITCS)*, Cambridge, MA, USA, pp. 309–325, 2012.
3. P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology – EUROCRYPT '99*, pp. 223–238, 1999.
4. S. Kadhe *et al.*, "FastSecAgg: Scalable secure aggregation for privacy-preserving federated learning," *arXiv preprint arXiv:2009.11248*, 2020.
5. C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.
6. K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. 2017 ACM SIGSAC Conf. Computer and Communications Security (CCS)*, Dallas, TX, USA, pp. 1175–1191, 2017.
7. S. P. Kasiviswanathan *et al.*, "What can we learn privately?," *SIAM J. Comput.*, vol. 40, no. 3, pp. 793–826, 2011.
8. ARM Limited, "TrustZone technology for ARMv8-M architecture," ARM Technical White Paper, 2016.
9. J. W. Kim, D. H. Kim, and B. Jang, "Application of local differential privacy to indoor positioning data," *IEEE Access*, vol. 6, pp. 51555–51565, 2018.
10. Intel Corporation, "Intel Software Guard Extensions Programming Reference," Document No. 329298-002US, 2015.
11. O. Babalola, J. Cano, N. H. Tran *et al.*, "Survey of privacy preserving federated tree-based learning," *Artificial Intelligence Review*, vol. 59, no. 7, art. no. 123, 2026. (to appear)
12. O. Ohrimenko *et al.*, "Oblivious multi-party machine learning on trusted processors," in *Proc. 25th USENIX Security Symp. (USENIX Security 16)*, Austin, TX, USA, pp. 619–636, 2016.
13. N. Papernot *et al.*, "Scalable private learning with PATE," in *Proc. 6th Int. Conf. Learning Representations (ICLR)*, Vancouver, BC, Canada, 2018.
14. S. H. Hashemi *et al.*, "Secure genomic data analysis using Intel SGX and homomorphic encryption," *BMC Bioinformatics*, vol. 21, no. 1, art. no. 449, 2020.
15. R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd IEEE Symp. Foundations of Computer Science (FOCS)*, pp. 136–145, 2001.
16. F. Loukil, C. Ghedira-Guegan, K. Boukadi, and A.-N. Benharkat, "Privacy-preserving IoT data aggregation based on blockchain and homomorphic encryption," *Sensors*, vol. 21, no. 7, art. no. 2452, 2021.
17. R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, pp. 2118–2125, 2018.
18. F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Čapkun, and A.-R. Sadeghi, "Software grand exposure: SGX cache attacks are practical," in *Proc. 11th USENIX Workshop Offensive Technol. (WOOT)*, Vancouver, BC, Canada, 2017.
19. A. Nejadhossein Qasemabadi, S. Mozaffari, M. Ahmadi, and S. Alirezaee, "Privacy-preserving lane change prediction using recurrent neural network with secure multiparty computation," *Int. J. Intell. Transp. Syst. Res.*, vol. 23, pp. 893–907, 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Publisher and/or the editor(s). Publisher and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.