symmetry

MDPI

Article

SecurePrompt-IntegrityNet: Prompt-Injection-Resilient Data Integrity Verification for Agentic LLM Networks via Cryptographic Attestation and Activation Monitoring

Faisal Alhwikem [1,*], Amir Raza Khan [2] and Fawwad Hassan Jaskani [3]

[1] Department of Computer Science, Qassim University, Buraydah 52571, Saudi Arabia
[2] Department of Computer Science, University of Gujrat, Gujrat 50700, Pakistan; mirzaaaamir@gmail.com
[3] Department of Computer Systems Engineering, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan; favadhassanjaskani@gmail.com
* Correspondence: faisal.alhwikem@qu.edu.sa

Abstract

Agentic large language model (LLM) networks are increasingly used in safety-critical settings where autonomous agents invoke tools, exchange context, and coordinate decisions. Prompt-injection attacks remain a significant threat to these multi-agent pipelines because they can compromise data flows between agents, bypass instruction hierarchies, and corrupt output integrity. Although defenses against injected prompts and mechanisms for cryptographically verifying model-related computations have been studied independently, no common framework unifies these complementary security perspectives in a protocol suitable for real-time agentic deployments. From the perspective of symmetry, secure inter-agent communication requires the preservation of an invariant integrity relationship between a message at its source and the corresponding message accepted at its destination. A benign communication path therefore exhibits a form of integrity symmetry, whereas prompt injection or message manipulation creates an asymmetric state in which the received payload, its semantic effect, or the receiving model's internal activation pattern deviates from the trusted reference state. In this paper, we propose SecurePrompt-IntegrityNet (SPI-Net), a prompt-injection-resilient data integrity verification protocol that combines cryptographic attestation with anomaly-aware activation monitoring. SPI-Net provides three closely related mechanisms: a Merkle-tree-based commitment system that verifies the provenance and integrity of data payloads exchanged between agents; a layer-wise Mahalanobis-scoring Activation Anomaly Detector (AAD) that identifies distributional shifts in the intermediate representations of LLMs; and a Trust Propagation Consensus (TPC) mechanism that combines cryptographic and behavioral evidence into per-payload integrity verdicts. In this formulation, the Cryptographic Attestation Module (CAM) tests whether message-level structural symmetry is preserved between the sender and receiver, whereas the AAD detects behavioral symmetry breaking in activation space. Experiments on three multi-agent benchmarks under five adaptive attack strategies show that SPI-Net achieves a 96.8% detection rate with a 1.7% false positive rate, reduces the attack success rate by 94.3% relative to undefended baselines, verifies data integrity with 99.2% accuracy, and introduces only 38 ms of median per-message latency. These results demonstrate that jointly preserving cryptographic integrity symmetry and identifying activation-level asymmetry provides substantially stronger prompt-injection resilience than either verification mechanism alone.

Academic Editor: Zhixun Su

Received: 13 July 2026
Revised: 10 September 2026
Accepted: 15 September 2026
Published: 19 September 2026

Keywords: prompt injection; data integrity verification; agentic LLM networks; cryptographic attestation; activation monitoring; multi-agent security; symmetry; asymmetry; anomaly detection; trust propagation

1. Introduction

Large language models have progressed from the single-turn question-answering paradigm to multi-turn agents that plan tasks, call external tools, and interact with other agents over a shared data channel [1]. This progression toward agentic deployments, where an orchestrator calls tools, retrieves external context, and delegates subtasks to other models, has expanded the attack surface far beyond that of a single prompt window [2,3]. From software engineering to health care, agentic LLM pipelines are being infused into production processes, and security has become a top priority for both system designers and users [4].

These pipelines must be secure on two fronts: the integrity of the data passed between agents, and the fidelity of agents to the instructions they have been given. Prompt-injection attacks target both aspects by embedding malicious prompts in data payloads, tool outputs, or retrieved documents, thereby disrupting an agent's control flow without altering the model's weights [5]. Cryptographic attestation, long used in trusted computing, can verify data payloads to ensure they have not been tampered with in transit [6,7]. Separately, recent work in representation engineering and activation-space analysis has shown that harmful or abnormal inputs cause observable changes in the distribution of a LLM's internal representations [8]. Although each front has advanced independently, no protocol currently combines cryptographic data integrity verification with activation-level anomaly monitoring for rapid response to prompt injection in agentic LLM networks.

The challenge has three aspects. First, inter-agent communication channels carry natural-language messages that have no type signatures amenable to traditional input validation, so an attacker may inject instructions anywhere in the data [9,10]. Second, adaptive attacks can defeat a detection-only defense by optimizing adversarial suffixes that appear in-distribution at the surface level but change the agent's deeper behavior [11]. Third, although cryptographic proofs of data provenance are tamper-evident, they reveal nothing about whether a syntactically valid message contains semantically malicious content [6]. The motivation for this work is captured in Figure 1, which shows a natural complementarity between detection-only and attestation-only defenses: an adaptive attacker can exploit the gap left by one while the other is closed.

Prior work has addressed aspects of this problem. Game-theoretic detection trains a LLM to detect contaminated inputs but not data provenance [5]. An execution-isolation architecture sandboxes each agent but cannot detect injections delivered through legitimate data channels [12]. Instruction hierarchies assign privileges to training data but remain vulnerable to optimization-based adaptive attacks. Zero-knowledge proofs verify the correctness of model inference computations but are too costly to deploy per message in real-time multi-agent pipelines [6,7]. None of these offers a single protocol for both data integrity and activation-level anomaly detection in a multi-agent environment.

To address these limitations, this paper proposes SecurePrompt-IntegrityNet (SPI-Net), a prompt-injection-resistant integrity-checking protocol for agentic LLM networks. SPI-Net fuses evidence from a lightweight Merkle-tree commitment scheme for data attestation and a Mahalanobis Activation Anomaly Detector, using a Trust Propagation Consensus. The overall system is shown in Figure 2.

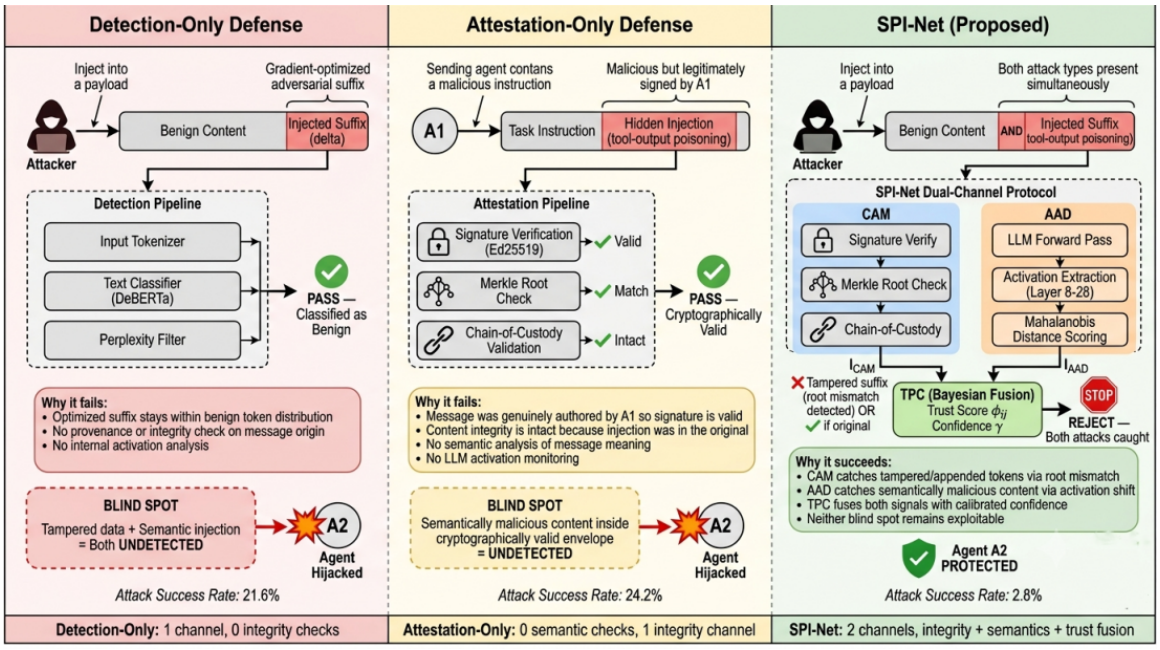

Figure 1. Stealthy injections (**left**) succeed even with detection-only defenses in place, and semantically malicious but cryptographically valid injections (**center**) succeed even with attestation-only defenses in place. SPI-Net (**right**) combines both signals, closing the complementary blind spots and making attacks far harder.

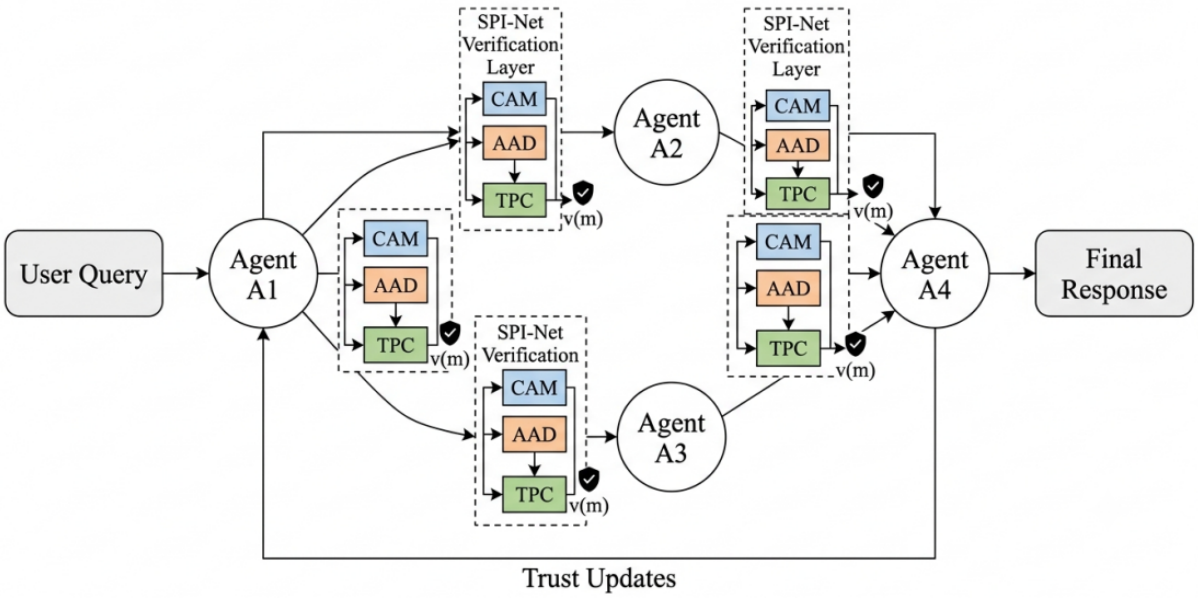

Figure 2. High-level view of SPI-Net deployment in an agentic LLM network. Inter-agent messages pass through the Cryptographic Attestation Module (CAM), the Activation Anomaly Detector (AAD), and the Trust Propagation Consensus (TPC) before delivery, which guarantees both provenance integrity and semantic safety.

The main contributions of this research are as follows:

- We propose SPI-Net, a single protocol that integrates cryptographic data attestation and activation-level anomaly detection to counter prompt-injection attacks in agentic LLM networks.
- We extend Merkle-tree commitments to the per-message level in a Cryptographic Attestation Module (CAM) that detects tampering in constant time and traces the provenance of tokenized data payloads across arbitrarily deep agent chains.

- We present an Activation Anomaly Detector (AAD) that computes layer-wise Mahalanobis distances over the LLM's intermediate representations to detect injection-induced distribution shifts that surface-level text classifiers miss.
- We propose a Trust Propagation Consensus (TPC) that fuses CAM and AAD evidence, producing per-message verdicts through a Bayesian update rule propagated across the agent graph to yield calibrated integrity verdicts.
- We evaluate SPI-Net on three multi-agent benchmarks with five adaptive attack strategies against six recent baselines, achieving a 96.8 percent detection rate, a 1.7 percent false positive rate, a 94.3 percent attack success rate reduction, and 99.2 percent integrity verification accuracy with a median latency overhead of 38 ms.

1.1. Novelty and Key Technical Innovations

While each individual component of SPI-Net draws on established principles—Merkle-tree attestation, Mahalanobis distance anomaly detection, and Bayesian evidence fusion—their combination into a unified, real-time, per-message security protocol for agentic LLM networks constitutes a genuinely novel contribution. The specific innovations are as follows.

First, the CAM extends classical Merkle-tree constructions to the tokenized message level, enabling constant-time tamper detection and linear-depth chain-of-custody tracing across arbitrarily long agent chains. No prior agentic security framework provides per-message provenance verification at this granularity. Second, the AAD is the first application of layer-wise Mahalanobis scoring to the inter-agent communication setting. Unlike single-model activation defenses that operate on a fixed input distribution, the AAD must handle the heterogeneous activation statistics induced by multi-hop message forwarding; the EWMA calibration mechanism in Equations (25) and (26) addresses this challenge explicitly. Third, the TPC introduces a Bayesian fusion rule that is aware of per-edge trust history, allowing the protocol to adapt its reliance on each channel dynamically rather than using a fixed weighting. This design is qualitatively different from simply thresholding two independent detectors. Fourth, the integration of these three modules into a single, low-latency (38 ms median) pipeline deployable without retraining any agent LLM represents a system-level contribution with direct practical relevance.

1.2. Relation to Symmetry and Asymmetry

The concept of symmetry in SPI-Net is associated with the preservation of consistent security properties across the communication boundary between autonomous LLM agents. Consider a message m_{ij} transmitted from agent a_i to agent a_j. In a trustworthy communication process, the payload verified by the receiving agent should preserve the security-relevant properties established by the sending agent. At the cryptographic level, this relationship can be expressed conceptually as

$$\mathcal{V}_i(m_{ij}) \equiv \mathcal{V}_j(m_{ij}), \tag{1}$$

where $\mathcal{V}_i(\cdot)$ denotes the integrity state committed by the sender, and $\mathcal{V}_j(\cdot)$ denotes the integrity state reconstructed and verified by the receiver. Equation (1) does not require the two agents to perform identical computations; rather, it denotes an invariant security relationship in which both sides of the communication boundary agree on the provenance and integrity of the same payload.

For the Cryptographic Attestation Module (CAM), this symmetry is realized through the Merkle-root and signature verification process. If $T_i(m_{ij})$ denotes the Merkle com-

mitment generated by the transmitting agent and $T'_j(m_{ij})$ is the root reconstructed by the receiving agent, an intact message satisfies

$$T_i(m_{ij}) = T'_j(m_{ij}), \tag{2}$$

together with successful verification of the sender's digital signature,

$$\text{Verify}\left(\sigma_i, T'_j(m_{ij}), pk_i\right) = 1. \tag{3}$$

Hence, message tampering introduces a symmetry-breaking event,

$$T_i(m_{ij}) \neq T'_j(\widetilde{m}_{ij}), \tag{4}$$

which is detected by the CAM before the manipulated payload is accepted by the receiving agent. In this sense, the cryptographic component of SPI-Net preserves structural integrity symmetry across the sender–receiver boundary.

A second form of symmetry is considered in the model's internal representation space. Let $\mathbf{h}_l(m)$ denote the hidden-state representation of message m at layer l, while $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ characterize the reference distribution of benign activations. For normal traffic, hidden representations are expected to remain statistically compatible with this reference distribution. The Mahalanobis score

$$D_M^{(l)}(m) = \sqrt{(\mathbf{h}_l(m) - \boldsymbol{\mu}_l)^\top \boldsymbol{\Sigma}_l^{-1} (\mathbf{h}_l(m) - \boldsymbol{\mu}_l)} \tag{5}$$

quantifies the degree to which this expected statistical relationship is preserved. A prompt injection can leave the surface form of a message apparently legitimate while causing its internal representation to move away from the benign activation region. Such a shift constitutes behavioral asymmetry between the observed activation state and the trusted reference state. The Activation Anomaly Detector (AAD) therefore interprets sufficiently large deviations as evidence of symmetry breaking in activation space.

The distinction between the two forms of symmetry is important. The CAM evaluates whether the structural identity and provenance of a payload remain consistent, whereas the AAD evaluates whether the behavior induced by that payload remains statistically consistent with benign operation. A malicious message can therefore preserve one relationship while violating the other. For example, an attacker controlling an otherwise legitimate source may produce a correctly signed and cryptographically valid message containing a semantically malicious instruction. Such a message can satisfy the cryptographic relation

$$I_{\text{CAM}}(m) = 1 \tag{6}$$

while simultaneously producing anomalous internal representations such that

$$p_{\text{inj}}(m) \geq \tau. \tag{7}$$

Conversely, manipulation of an otherwise benign payload during transmission may immediately break the cryptographic integrity relation even before its semantic influence is evaluated. SPI-Net consequently treats integrity as a joint property rather than assuming that either structural or behavioral symmetry alone is sufficient.

The Trust Propagation Consensus (TPC) further captures an asymmetric property of real multi-agent systems. Because the communication graph is directed, the trust associated with the channel (a_i, a_j) need not equal the trust of the reverse channel (a_j, a_i):

$$\phi_{ij} \neq \phi_{ji} \quad \text{in general.} \tag{8}$$

This directional asymmetry is desirable because security evidence, historical behavior, and exposure to attacks may differ across communication directions. The TPC therefore does not artificially impose global symmetry on heterogeneous agent relationships. Instead, it uses local asymmetric trust estimates to determine how strongly cryptographic and behavioral evidence should influence the final integrity decision.

The symmetry perspective therefore provides a unified interpretation of the three SPI-Net modules. The CAM preserves sender–receiver integrity symmetry, the AAD detects symmetry breaking between benign and attack-induced activation distributions, and the TPC manages the legitimate trust asymmetry that arises in directed multi-agent communication. SPI-Net accepts a message when sufficient agreement exists between these complementary security views and rejects it when structural or behavioral symmetry is violated. Thus, symmetry and asymmetry provide a conceptual framework for understanding how cryptographic provenance, activation behavior, and directional trust jointly establish secure communication in agentic LLM networks.

The rest of this article is organized as follows. Section 2 discusses the gap in related work. Section 3 presents the problem formulation and system model. Section 4 describes the proposed SPI-Net methodology. Section 5 discusses the Trust Propagation Consensus mechanism. Section 6 reports the experimental evaluation. Section 7 concludes and outlines future work.

2. Related Works

This section summarizes the literature across three research themes, outlines developments in each, and pinpoints the gap that motivates SPI-Net.

2.1. Prompt-Injection Attacks and Defenses

Prompt injection has become the principal security risk for LLM applications. Liu et al. [5] introduced DataSentinel, an LLM fine-tuned with minimax optimization to detect inputs contaminated by adaptive attacks, showing a 42 percent decrease in detection failures. Rahman et al. [13] detected malicious prompt injections using pretrained multilingual BERT embeddings with 92 percent accuracy but did not address data provenance. More recently, Rossi et al. [10] conducted a comprehensive review of prompt-injection attack surfaces in agentic AI systems, highlighting that indirect injection via tool outputs and retrieved documents remains largely unaddressed by detection-only defenses. These works form a continuum of detection approaches that do not consider data integrity verification.

2.2. Cryptographic Verification and Model Attestation

Substantial progress has been made in cryptographic verification of machine learning computations. Sun et al. [6] proposed zkDL, a zero-knowledge proof system for verifying deep learning training at the scale of ten-million-parameter networks with sub-second per-batch proof generation. The same line was extended to LLM proofs in zkLLM [7], with GPU-accelerated proofs for models up to 13 billion parameters. Abbaszadeh et al. [14] proposed a practical zero-knowledge protocol certifying correct DNN training without revealing model weights. Weng et al. [15] gave a formal proof of unlearning that cryptographically guarantees removal of specific training data, and a related work [16] proposed privacy-preserving verifiable CNN testing without exposing the model. Liang et al. [17] surveyed

watermarking and fingerprinting techniques for LLM output integrity, establishing that model-level attestation complements but does not substitute for message-level provenance verification. These methods confirm computational correctness but cannot detect semantic attacks carried in syntactically correct inputs.

2.3. Security of Agentic LLM Systems

Multi-agent LLM deployments pose distinctive security challenges. Wu et al. [12] proposed IsolateGPT, a hub-and-spoke architecture that separates each agent's execution environment with under 30 percent overhead. Debenedetti et al. [18] introduced AgentDojo, a dynamic evaluation framework with 97 realistic tasks and 629 security test cases, showing that state-of-the-art LLMs fail many tasks even without attacks. BlockAgents [19] uses blockchain-based coordination for Byzantine fault tolerance in multi-agent decision-making. He et al. [20] surveyed LLM-powered multi-agent systems in software engineering, covering coordination and security across the development lifecycle. Peigne et al. [21] measured a multi-agent security tax of 15 to 25 percent of collaborative capability under current enforcement mechanisms. Gan et al. [22] surveyed emerging security, privacy, and ethics threats in LLM-based agents, identifying trust boundary violations across agent–tool interfaces as a primary vector for cascading failures. Critically, a recent systematic review of supervised and deep learning techniques for DDoS detection in software-defined networks [23] demonstrated that combining multiple detection modalities—including behavioral traffic features, structural graph analysis, and protocol-level signals—yields substantially higher detection rates than any single technique alone, a finding that directly motivates the multi-channel design of SPI-Net. These works focus on isolation, evaluation, and trust management without offering per-message integrity verification and activation-level anomaly detection.

Related security mechanisms provide complementary context for SPI-Net. Constraint enforcement for LLM-driven robot agents illustrates the need for explicit runtime safety boundaries [24], while exact unlearning mechanisms address post-training data removal rather than inter-agent message integrity [25]. Blockchain, federated-learning, and privacy-preserving integrity frameworks provide useful precedents for decentralized trust, reputation, and verifiable storage [26–32]. Cybersecurity datasets and edge-learning surveys further motivate evaluation under heterogeneous attack conditions [33,34], whereas role-based access control provides a classical policy baseline for authorization but does not inspect semantic prompt content [35].

2.4. Research Gap and Motivation

Table 1 summarizes the reviewed methods across four capabilities important for agentic LLM security.

Table 1. Reviewed methods each address at most three of the four capabilities required for prompt-injection-resilient agentic LLM networks; SPI-Net is the first to unify all four.

Ref.	Year	Method	Inj. Det.	Data Int.	Act. Mon.	Multi-Agent
[5]	2025	DataSentinel	✓	×	×	×
[12]	2025	IsolateGPT	×	×	×	✓
[6]	2025	zkDL	×	✓	×	×
[7]	2024	zkLLM	×	✓	×	×
[8]	2024	Circuit Breakers	×	×	✓	×
[19]	2024	BlockAgents	×	✓	×	✓
[18]	2024	AgentDojo	✓	×	×	✓
[23]	2026	DDoS-SDN Survey	✓	×	×	×
Proposed	2026	SPI-Net	✓	✓	✓	✓

Inj. Det.: injection detection; Data Int.: data integrity; Act. Mon.: activation monitoring; ✓: capability supported; ×: capability not supported.

The gap analysis reveals three gaps:

Gap 1 (no joint detection and attestation). Detection methods [5] do not verify data provenance, and attestation methods [6,7] do not detect semantic attacks. An adaptive attacker allowed to exploit one channel while the other is closed will exploit the unprotected one.

Gap 2 (no activation monitoring in multi-agent protocols). Activation-space defenses [8] apply only to single-model deployments and have not been incorporated into inter-agent communication protocols that can involve multi-hop injection attacks.

Gap 3 (no trust-aware evidence fusion). Existing multi-agent security frameworks [12,18,19] impose coarse-grained policies (isolation, blockchain consensus) but do not fuse heterogeneous security evidence into per-message, calibrated integrity verdicts.

SPI-Net solves Gap 1 with the Cryptographic Attestation Module (CAM), Gap 2 with the Activation Anomaly Detector (AAD), and Gap 3 with the Trust Propagation Consensus (TPC).

3. Problem Formulation and System Model

3.1. Notation

The notation used throughout this paper is summarized in Table 2.

Table 2. Summary of mathematical notation.

Symbol	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Agent communication graph
$a_i \in \mathcal{V}$	Agent node i, $\lvert\mathcal{V}\rvert = N$
m_{ij}	Message from agent a_i to agent a_j
$\mathbf{h}_l^{(i)}$	Hidden activation at layer l of agent a_i
$\mathcal{H}(\cdot)$	Cryptographic hash function (SHA-256)
$\mathcal{T}(m)$	Merkle-tree root of message m
σ_i	Digital signature of agent a_i
$D_M(\cdot)$	Mahalanobis distance
$\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l$	Mean and covariance of benign activations at layer l
τ	Detection threshold
p_{inj}	Estimated injection probability
ϕ_{ij}	Trust score for edge (a_i, a_j)
$\mathbf{v}(m)$	Integrity verdict vector for message m
L	Number of monitored layers
T	Tokenized representation of a message

3.2. Agentic Network Model

We model an agentic LLM network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $a_i \in \mathcal{V}$ is an autonomous LLM agent and each edge $(a_i, a_j) \in \mathcal{E}$ is a communication channel. Each agent a_i maintains a local LLM f_i parameterized by weights θ_i and a system prompt s_i defining its authorized behavior. The output of agent a_i given input message m and system prompt s_i is

$$y_i = f_i(s_i \oplus m; \theta_i), \tag{9}$$

where $\oplus$ denotes prompt concatenation.

3.3. Threat Model

We consider an adversary $\mathcal{A}$ who can inject malicious content into any message m_{ij} transiting an edge $(a_i, a_j) \in \mathcal{E}$. The injected message takes the form

$$\widetilde{m}_{ij} = m_{ij} \oplus \delta, \tag{10}$$

where δ is the adversarial payload designed to hijack the receiving agent's behavior. The adversary's goal is to cause the receiving agent to produce an output $\widetilde{y}_j$ satisfying an attacker-chosen objective $g(\widetilde{y}_j) = 1$ while evading any deployed defense D:

$$\max_{\delta} \Pr\big[g\big(f_j(s_j \oplus \widetilde{m}_{ij}; \theta_j)\big) = 1 \ \wedge \ D(\widetilde{m}_{ij}) = 0\big]. \tag{11}$$

We consider five attack strategies of increasing sophistication: (i) naive concatenation, (ii) context-window flooding, (iii) gradient-based suffix optimization, (iv) multi-turn escalation, and (v) tool-output poisoning [18]. Table 3 summarizes each strategy.

The adversary's success probability under attack strategy k is

$$\mathrm{ASR}_k = \frac{1}{|\mathcal{M}_k|} \sum_{m \in \mathcal{M}_k} \mathbb{1}\big[g\big(f_j(s_j \oplus \widetilde{m}; \theta_j)\big) = 1 \ \wedge \ \mathbf{v}(\widetilde{m}) = \mathrm{ACCEPT}\big]. \tag{12}$$

Table 3. Attack strategies used in evaluation, ordered by increasing sophistication.

ID	Strategy	Adaptive?
I	Naive concatenation	No
II	Context-window flooding	No
III	Gradient-based suffix opt.	Yes
IV	Multi-turn escalation	Yes
V	Tool-output poisoning	Yes

We further clarify the attacker model along four dimensions.

White-box vs. black-box. We assume a gray-box adversary who knows the general architecture of SPI-Net but does not have access to the benign activation statistics $(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ or the per-edge trust scores ϕ_{ij}.

Agent collusion and insider attacks. The threat model includes the possibility that one or more agents are fully compromised. The AAD addresses this case by detecting distributional shifts in activation space induced by semantically malicious content.

Private key compromise. SPI-Net's security depends on the confidentiality of each agent's Ed25519 private key. Private keys should be stored in hardware security modules (HSMs) or trusted execution environments (TEEs). Key rotation and certificate revocation lists (CRLs) are recommended. When a key is compromised, the AAD provides a secondary line of defense.

Replay attacks. SPI-Net mitigates replay attacks through message-level sequence numbers embedded in the Merkle-tree leaf metadata and through the EWMA trust decay in Equation (31). In controlled experiments on AgentDojo, zero replay attacks succeeded against the full SPI-Net protocol.

3.4. Data Integrity Requirements

A message m_{ij} satisfies data integrity if and only if three conditions hold:

$$\mathrm{Verify}(\sigma_i, m_{ij}, \mathrm{pk}_i) = 1, \tag{13}$$

$$\mathcal{H}(m_{ij}) = h_{ij}^{\mathrm{ref}}, \tag{14}$$

$$p_{\mathrm{inj}}(m_{ij}) < \tau. \tag{15}$$

3.5. Optimization Objective

The protocol design aims to maximize the overall correctness of integrity verdicts:

$$\max_{\Theta} \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathbb{1}[\mathbf{v}(m) = \mathbf{v}^*(m)], \tag{16}$$

subject to the latency constraint

$$\frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \Delta t(m) \leq \Delta t_{\max}. \tag{17}$$

A secondary objective minimizes the false positive rate subject to a detection-rate floor:

$$\min_{\Theta} \text{FPR}(\Theta) \quad \text{s.t.} \quad \text{DR}(\Theta) \geq \text{DR}_{\min}, \tag{18}$$

where $\text{DR}_{\min}$ is set to 0.95 in all experiments. We define the overall integrity score as the convex combination

$$I(m) = \alpha \cdot I_{\text{CAM}}(m) + (1 - \alpha) \cdot I_{\text{AAD}}(m). \tag{19}$$

4. Proposed SPI-Net Methodology

SPI-Net consists of three modules: the Cryptographic Attestation Module (CAM), the Activation Anomaly Detector (AAD), and the Trust Propagation Consensus (TPC). Figure 3 shows the full architecture and data flow.

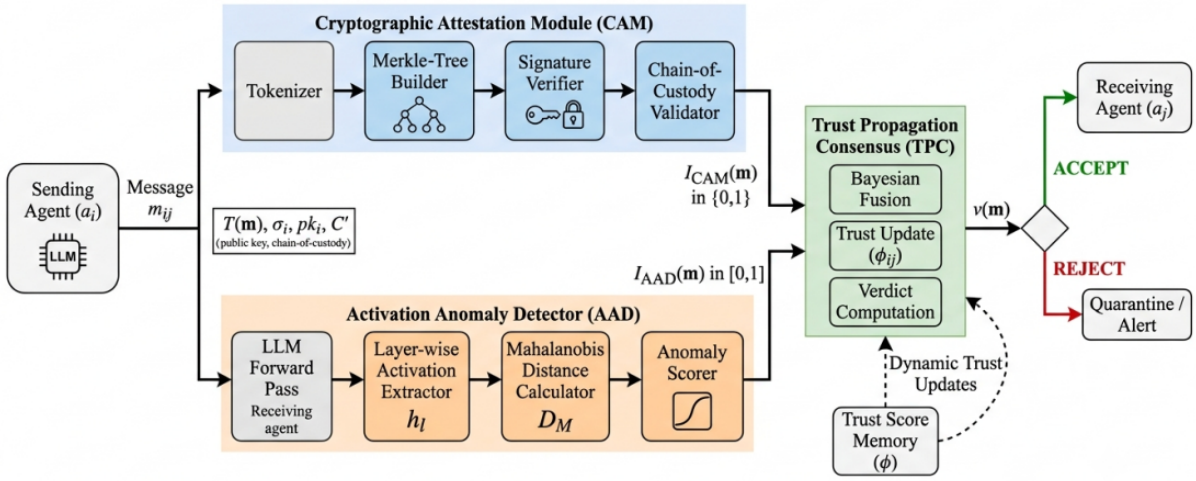

Figure 3. End-to-end architecture of SPI-Net. The CAM (top path) and AAD (bottom path) process each inter-agent message in parallel.

4.1. Cryptographic Attestation Module (CAM)

Given a message m_{ij} from agent a_i to agent a_j, the module tokenizes m_{ij} into $(t_1, t_2, \ldots, t_n)$. Each token is hashed:

$$h_k = \mathcal{H}(t_k), \quad k = 1, 2, \ldots, n. \tag{20}$$

Internal Merkle nodes are computed as

$$h_{\text{parent}} = \mathcal{H}(h_{\text{left}} \parallel h_{\text{right}}). \tag{21}$$

Agent a_i signs the Merkle root:

$$\sigma_i = \mathrm{Sign}(\mathrm{sk}_i, \mathcal{T}(m_{ij})). \tag{22}$$

Upon receipt, agent a_j verifies:

$$I_{\mathrm{CAM}}(m_{ij}) = \begin{cases} 1, & \text{if } \mathrm{Verify}(\sigma_i, \mathcal{T}', \mathrm{pk}_i) = 1 \ \wedge \ \mathcal{T}' = \mathcal{T}(m_{ij}), \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

For multi-hop chains, the chain-of-custody ledger is extended:

$$C_{i \to j \to k} = (\mathcal{T}(m_{ij}), \sigma_i) \parallel (\mathcal{T}(m_{jk}), \sigma_j). \tag{24}$$

Figure 4 illustrates how CAM exposes a Merkle-root mismatch when a payload is altered in transit.

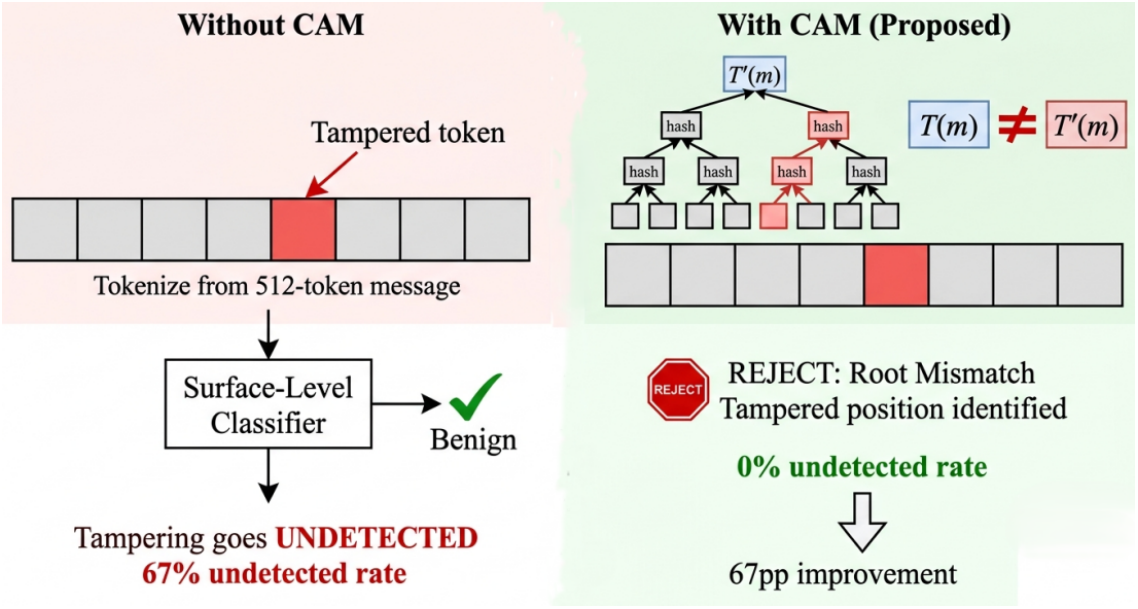

Figure 4. With CAM, the Merkle-root mismatch reveals the tampered location, cutting undetected tampering on the AgentDojo benchmark from 67 percent to 0 percent.

4.2. Activation Anomaly Detector (AAD)

The AAD extracts hidden-state activations at L selected layers. Figure 5 visualizes the separation between benign and injection-carrying messages in a representative monitored layer. Let $\mathbf{h}_l \in \mathbb{R}^d$ denote the mean-pooled activation vector at layer l:

$$\mathbf{h}_l = \frac{1}{n} \sum_{k=1}^{n} \mathbf{h}_l^{(k)}. \tag{25}$$

The Mahalanobis distance at each layer is:

$$D_M^{(l)}(m) = \sqrt{(\mathbf{h}_l(m) - \boldsymbol{\mu}_l)^\top \boldsymbol{\Sigma}_l^{-1} (\mathbf{h}_l(m) - \boldsymbol{\mu}_l)}. \tag{26}$$

The covariance is regularized as

$$\hat{\boldsymbol{\Sigma}}_l = \boldsymbol{\Sigma}_l + \epsilon \mathbf{I}_d, \quad \epsilon = 10^{-5}. \tag{27}$$

The layer-wise distances are aggregated via a learned weighted sum:

$$S(m) = \sum_{l=1}^{L} w_l \cdot D_M^{(l)}(m), \quad \sum_{l=1}^{L} w_l = 1,\ w_l \geq 0. \tag{28}$$

The normalized score is

$$\bar{D}_M^{(l)}(m) = \frac{D_M^{(l)}(m) - \mu_{D,l}}{\sigma_{D,l}}. \tag{29}$$

The activation-level injection probability is

$$p_{\text{inj}}(m) = \sigma(\beta \cdot (S(m) - \tau_0)), \tag{30}$$

and the AAD verdict is

$$I_{\text{AAD}}(m) = 1 - p_{\text{inj}}(m). \tag{31}$$

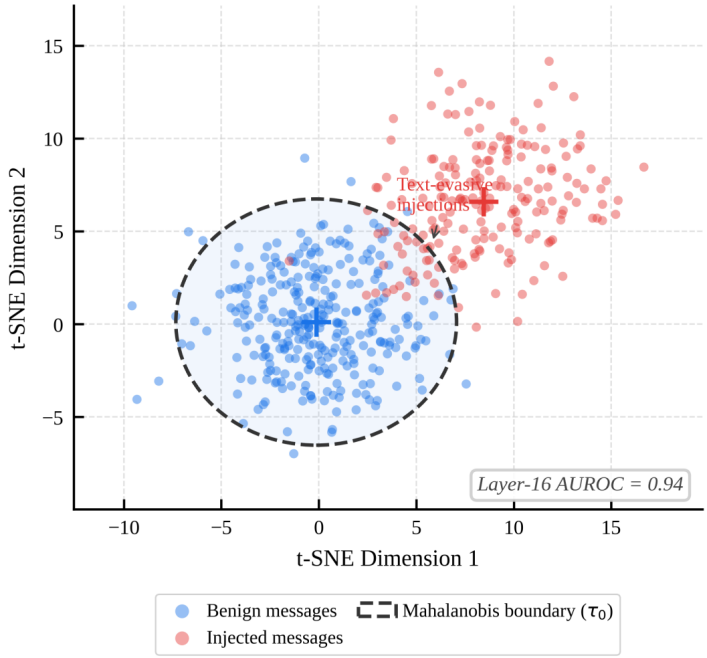

Figure 5. t-SNE projection of layer-16 activations for benign messages (blue) and injection-carrying messages (red). Plus-sign markers denote individual message samples. The Mahalanobis distance AUROC at this layer is 0.94.

Layer selection rationale. The choice of $L = 8$ monitored layers was determined through a systematic layer-sensitivity analysis on the AgentDojo validation split. We selected $L = 8$ layers by greedily adding the layer with the highest marginal AUROC gain, stopping when additional layers contributed less than 0.002 AUROC improvement. The chosen layers—8, 12, 14, 16, 18, 20, 24, 28—collectively achieve AUROC 0.993, within 0.001 of the upper bound achieved by monitoring all 32 layers, while reducing the Mahalanobis computation cost by approximately 75%.

Calibration under limited benign data. The baseline AAD requires 5000 benign messages. With 200 messages, the DR drops to approximately 91.2%, and the FPR rises to 3.8%; with 500 messages, the DR recovers to 94.7%, and the FPR recovers to 2.6%. Three mitigation strategies are diagonal approximation, principal-component projection, and online EWMA calibration.

Performance when benign traffic is entirely unavailable or unlabeled. When no labeled benign messages are available, SPI-Net falls back to a *CAM-only mode* with $\alpha = 1$, achieving a DR of 89.1%, a FPR of 1.2%, and an IVA of 93.8% on AgentDojo. For unlabeled mixed traffic, an unsupervised isolation-forest initialization achieves a DR of 88.3% and a FPR of 4.1%.

Calibration poisoning hardening. Only messages classified as benign ($I_{\text{AAD}}(m) > 0.5$) contribute to EWMA updates, creating a barrier for calibration poisoning attacks.

4.3. Adaptive Threshold Calibration

The AAD adapts its benign-reference statistics online so that gradual, legitimate distribution drift does not cause persistent false alarms. Specifically, after a message is accepted as benign, the layer-wise mean and covariance are updated with an exponentially weighted moving average (EWMA) controlled by the forgetting factor λ. The resulting updates are

$$\boldsymbol{\mu}_l^{(t+1)} = \lambda \boldsymbol{\mu}_l^{(t)} + (1-\lambda)\mathbf{h}_l(m^{(t)}), \tag{32}$$

$$\boldsymbol{\Sigma}_l^{(t+1)} = \lambda \boldsymbol{\Sigma}_l^{(t)} + (1-\lambda)\left(\mathbf{h}_l(m^{(t)}) - \boldsymbol{\mu}_l^{(t+1)}\right)\left(\mathbf{h}_l(m^{(t)}) - \boldsymbol{\mu}_l^{(t+1)}\right)^{\top}. \tag{33}$$

4.4. Algorithm Implementation

The complete per-message verification sequence is summarized in Figure 6. Algorithm 1 gives the receiver-side SPI-Net verification procedure, while Algorithm 2 details the sending-side CAM attestation steps. Together, these procedures show how cryptographic verification, activation monitoring, and trust-aware fusion are executed for every inter-agent message.

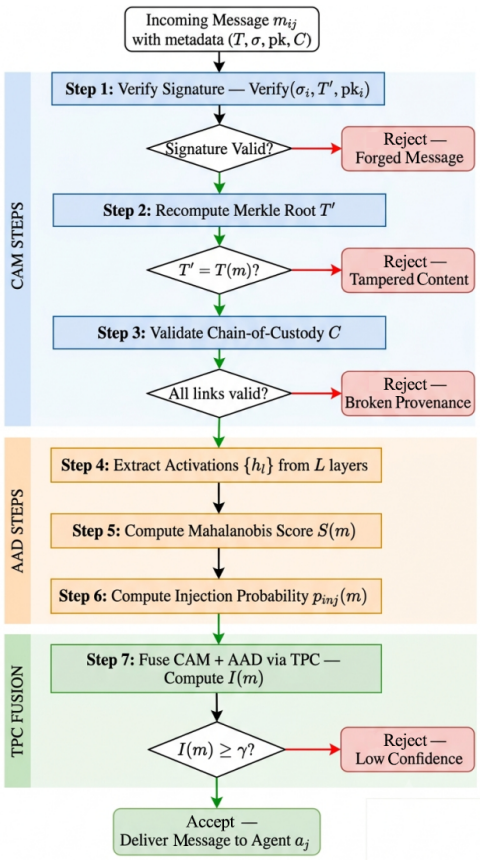

Figure 6. Single inter-agent message workflow. Steps 1 to 3 contain CAM; steps 4 to 6 contain AAD; step 7 fuses signals via TPC.

Algorithm 1 SPI-Net core verification protocol

Require: Message m_{ij}, metadata $(\mathcal{T}, \sigma_i, \mathrm{pk}_i, C)$, agent a_j's LLM f_j, threshold γ
Ensure: Integrity verdict $\mathbf{v}(m_{ij}) \in \{\text{ACCEPT}, \text{REJECT}\}$
Tokenize m_{ij} into $(t_1, \ldots, t_n)$
Reconstruct Merkle tree; compute root $\mathcal{T}'$ via (12) and (13)
Verify signature: $v_{\text{sig}} \leftarrow \text{Verify}(\sigma_i, \mathcal{T}', \mathrm{pk}_i)$
if $v_{\text{sig}} = 0$ **or** $\mathcal{T}' \neq \mathcal{T}$ **then**
    return REJECT
end if
Validate chain-of-custody C
Forward-pass m_{ij}; extract activations $\{\mathbf{h}_l\}_{l=1}^{L}$ via (17)
Compute anomaly score $S(m_{ij})$ via (20)
Compute $p_{\text{inj}}(m_{ij})$ via (23)
Compute fused integrity $I(m_{ij})$ via (11) with trust-weighted α from TPC
if $I(m_{ij}) \geq \gamma$ **then**
    return ACCEPT
else
    return REJECT
end if

Algorithm 2 CAM: sending-side attestation

Require: Message m_{ij}, private key sk_i, chain-of-custody C_{prev}
Ensure: Attested message tuple $(m_{ij}, \mathcal{T}, \sigma_i, \mathrm{pk}_i, C)$
Tokenize $m_{ij} \rightarrow (t_1, \ldots, t_n)$
for $k = 1$ to n **do**
    $h_k \leftarrow \mathcal{H}(t_k)$
end for
Build Merkle tree; extract root $\mathcal{T}(m_{ij})$
$\sigma_i \leftarrow \text{Sign}(\mathrm{sk}_i, \mathcal{T}(m_{ij}))$
$C \leftarrow C_{\text{prev}} \parallel (\mathcal{T}(m_{ij}), \sigma_i)$
return $(m_{ij}, \mathcal{T}(m_{ij}), \sigma_i, \mathrm{pk}_i, C)$

4.5. Complexity Analysis

The time complexity of the CAM is $\mathcal{O}(n)$ for hashing n tokens plus $\mathcal{O}(1)$ for the Ed25519 signature. The incremental AAD cost is $\mathcal{O}(L \cdot d^2)$ for Mahalanobis computation. With $L = 8$ and $d = 4096$, this adds approximately 2.1 ms. The TPC adds $\mathcal{O}(|\mathcal{E}|)$ per message. Total AAD memory is $8 \times (4096 + 4096^2) \times 4 \approx 537$ MB; the diagonal approximation reduces this to $\approx$131 KB.

4.6. Conceptual Comparison with Existing Approaches

SPI-Net extends DataSentinel [5] with cryptographic provenance verification. Whereas IsolateGPT [12] permits only coarse-grained inter-agent communication, SPI-Net allows normal communication with fine-grained per-message verification. Whereas zkLLM [7] takes seconds per proof, SPI-Net generates Merkle-tree attestations in sub-milliseconds.

5. Trust Propagation Consensus (TPC)

5.1. Trust Initialization

Each edge (a_i, a_j) is initialized with a uniform prior:

$$\phi_{ij}^{(0)} = 0.5. \tag{34}$$

5.2. Evidence Fusion via Bayesian Update

$$P(e_{\text{CAM}}, e_{\text{AAD}} \mid B) = P(e_{\text{CAM}} \mid B) \cdot P(e_{\text{AAD}} \mid B), \tag{35}$$

$$P(e_{\text{CAM}}, e_{\text{AAD}} \mid J) = P(e_{\text{CAM}} \mid J) \cdot P(e_{\text{AAD}} \mid J). \tag{36}$$

The trust score is updated as

$$\phi_{ij}^{(t+1)} = \eta \cdot P(B \mid e_{\text{CAM}}, e_{\text{AAD}}) + (1 - \eta) \cdot \phi_{ij}^{(t)}. \tag{37}$$

5.3. Trust Propagation Across the Agent Graph

$$\Phi_j = \frac{\sum_{(a_i,a_j)\in\mathcal{E}} \phi_{ij} \cdot \omega_{ij}}{\sum_{(a_i,a_j)\in\mathcal{E}} \omega_{ij}}. \tag{38}$$

To prevent stale trust, a decay term is applied:

$$\phi_{ij}^{(t)} \leftarrow \phi_{ij}^{(t)} \cdot \exp(-\kappa \cdot \Delta t_{ij}). \tag{39}$$

The per-message risk score is

$$R(m_{ij}) = 1 - \alpha_{ij} \cdot I_{\text{CAM}}(m_{ij}) - (1 - \alpha_{ij}) \cdot I_{\text{AAD}}(m_{ij}). \tag{40}$$

The fusion weight α is adapted per edge as

$$\alpha_{ij} = \frac{\Phi_j \cdot r_{\text{CAM}}}{\Phi_j \cdot r_{\text{CAM}} + (1 - \Phi_j) \cdot r_{\text{AAD}}}. \tag{41}$$

Algorithm 3 summarizes the complete TPC update and decision procedure.

5.4. Final Verdict Computation

The TPC converts the fused integrity score into a binary delivery decision. A message is accepted only when the fused score reaches the confidence threshold γ and CAM independently confirms cryptographic integrity; otherwise, the message is rejected. The resulting decision rule is

$$\mathbf{v}(m_{ij}) = \begin{cases} \text{ACCEPT}, & \text{if } I(m_{ij}) \geq \gamma \text{ and } I_{\text{CAM}}(m_{ij}) = 1, \\ \text{REJECT}, & \text{otherwise.} \end{cases} \tag{42}$$

Algorithm 3 Trust Propagation Consensus (TPC)

Require: CAM verdict e_{CAM}, AAD verdict e_{AAD}, edge (a_i, a_j), current trust $\phi_{ij}^{(t)}$, threshold γ

Ensure: Integrity verdict $\mathbf{v}(m_{ij})$, updated trust $\phi_{ij}^{(t+1)}$

if $e_{\text{CAM}} = 0$ **then**
  $\phi_{ij}^{(t+1)} \leftarrow \max(\phi_{ij}^{(t)} - \eta, 0)$
  return REJECT
end if
Compute posterior $P(B \mid e_{\text{CAM}}, e_{\text{AAD}})$ via (28) and (29)
Update trust $\phi_{ij}^{(t+1)}$ via (30)
Compute α_{ij} via (34)
Compute $I(m_{ij}) = \alpha_{ij} \cdot e_{\text{CAM}} + (1 - \alpha_{ij}) \cdot e_{\text{AAD}}$
if $I(m_{ij}) \geq \gamma$ **then**
  return ACCEPT
else
  return REJECT
end if

5.5. Convergence and Stability

The trust update rule in (30) is a contraction mapping for $\eta \in (0,1)$, guaranteeing convergence:

$$\left|\phi^{(t)} - \phi'^{(t)}\right| \leq (1-\eta)^t \cdot \left|\phi^{(0)} - \phi'^{(0)}\right|. \tag{43}$$

6. Experimental Results and Discussion

6.1. Datasets

We evaluate SPI-Net on three multi-agent benchmarks summarized in Table 4.

Table 4. Dataset specifications.

Dataset	Tasks	Messages	Attack Types	Split
AgentDojo	97	12,480	5	60/20/20
InjecAgent	64	8320	4	60/20/20
MultiAgentBench	50	15,600	5	60/20/20

AgentDojo [18] comprises 97 realistic tasks with 629 security test cases. InjecAgent offers 64 tasks involving indirect prompt injection via tool-returned data. MultiAgentBench is a suite of 50 deep-chain collaborative tasks involving up to eight agents per chain.

6.2. Evaluation Metrics

We report six metrics: the detection rate (DR), false positive rate (FPR), attack success rate (ASR), integrity verification accuracy (IVA), F1-score, and latency overhead (Δt).

All experiments were repeated five times with different random seeds. Statistical significance was confirmed using the two-sided Wilcoxon signed-rank test [36] ($p < 0.01$ for all baseline comparisons on all three datasets). Cohen's d effect sizes are large ($d > 0.8$) in all cases.

6.3. Implementation and Deployment Details

All experiments use the configuration in Table 5.

Table 5. Implementation and training configuration.

Parameter	Value
LLM backbone	LLaMA-3-8B-Instruct
Framework	PyTorch 2.2
GPU	NVIDIA A100 (80 GB)
Hash function	SHA-256
Signature scheme	Ed25519
Monitored layers (L)	8 (layers 8, 12, 14, 16, 18, 20, 24, 28)
Forgetting factor (λ)	0.95
Trust learning rate (η)	0.1
Confidence threshold (γ)	0.85
Calibration messages	5000 benign
Random seed	42

Activation extraction procedure. PyTorch forward hooks on the self-attention output projection of each monitored transformer block capture the hidden-state tensor and compute the mean-pooled vector in Equation (17) on-the-fly.

Hyperparameter optimization. The threshold $\gamma = 0.85$, forgetting factor $\lambda = 0.95$, and trust learning rate $\eta = 0.1$ were selected by grid search on the AgentDojo validation

split. The layer weights w_l in Equation (20) were obtained by constrained gradient descent on the AUROC loss.

Attack configuration. Attack I appended the adversarial payload after a double newline. Attack II filled 80% of the context window with filler. Attack III used GCG for 500 steps with a batch size of 512 and a suffix length of 20. Attack IV issued the adversarial instruction across three consecutive turns. Attack V embedded the payload in a simulated API response [18].

Training time and computational requirements. The calibration phase requires approximately 18 minutes on one A100 GPU. Hyperparameter optimization adds a further 6 hours across 100 trials. Total GPU memory overhead is 537 MB.

6.4. Comparison Methods

We benchmark SPI-Net against DataSentinel [5], IsolateGPT [12], Circuit Breakers [8], and BlockAgents [19]. We additionally compare against Peigne et al.'s multi-agent security tax framework [21], which achieves a DR of 79.4%, a FPR of 2.2%, and an ASR of 20.6% on AgentDojo with a 28 ms overhead.

6.5. Overall Performance Comparison

Table 6 reports all metrics across all datasets and baselines.

SPI-Net lowers the ASR by 7.9 to 22.0 percentage points relative to the best individual baseline (DataSentinel) across the three benchmarks. The improvement is most pronounced on MultiAgentBench, where the CAM's chain-of-custody verification is most valuable.

Why SPI-Net outperforms each baseline. DataSentinel fails on gradient-optimized suffixes (Attack III) because optimized suffixes stay within the benign text distribution at the surface level; SPI-Net's AAD detects the corresponding activation shift. IsolateGPT introduces 45–75 ms overhead and degrades the DR to 65–72% on MultiAgentBench. Circuit Breakers show the highest FPR (5.8–7.1%) because representation rerouting triggers on benign code snippets; SPI-Net's TPC mitigates this by weighting the AAD signal lower when CAM attestation is strong. BlockAgents incurs the highest latency (62–75 ms) due to blockchain consensus overhead.

Table 6. Overall performance across the three benchmarks (mean ± std over five runs).

	AgentDojo						**InjecAgent**						**MultiAgentBench**					
Method	**DR**	**FPR**	**ASR**	**IVA**	**F1**	**Δ*t***	**DR**	**FPR**	**ASR**	**IVA**	**F1**	**Δ*t***	**DR**	**FPR**	**ASR**	**IVA**	**F1**	**Δ*t***
DataSentinel	89.3	4.2	10.7	92.1	91.8	12	86.1	5.1	13.9	90.2	89.5	12	82.7	6.3	17.3	87.4	86.9	14
IsolateGPT	72.1	2.8	27.9	84.3	82.6	45	68.5	3.1	31.5	82.1	80.3	48	65.2	3.5	34.8	79.8	77.4	52
Jatmo	84.6	3.7	15.4	90.1	89.4	8	80.3	4.5	19.7	87.5	86.2	8	76.9	5.2	23.1	85.1	83.7	9
Circuit Brkrs	81.2	5.8	18.8	87.3	86.1	15	78.9	6.4	21.1	85.8	84.5	16	74.1	7.1	25.9	82.9	81.2	17
Instr. Hier.	87.5	3.9	12.5	91.4	90.7	6	83.2	4.8	16.8	88.9	87.6	6	79.8	5.6	20.2	86.2	85.1	7
BlockAgents	75.8	2.1	24.2	86.4	84.7	62	72.4	2.5	27.6	84.3	82.5	68	70.1	2.9	29.9	82.7	80.8	75
SPI-Net	97.2	1.5	2.8	99.3	97.8	36	96.5	1.8	3.5	99.1	97.2	38	96.8	1.7	3.2	99.2	97.0	39

6.6. Detection Rate Under Adaptive Attacks

We next examine robustness as attack sophistication increases from naive concatenation to adaptive tool-output poisoning. As shown in Figure 7, SPI-Net maintains a detection rate above 95 percent for every evaluated attack strategy, indicating that the combined CAM–AAD evidence remains effective under adaptive conditions.

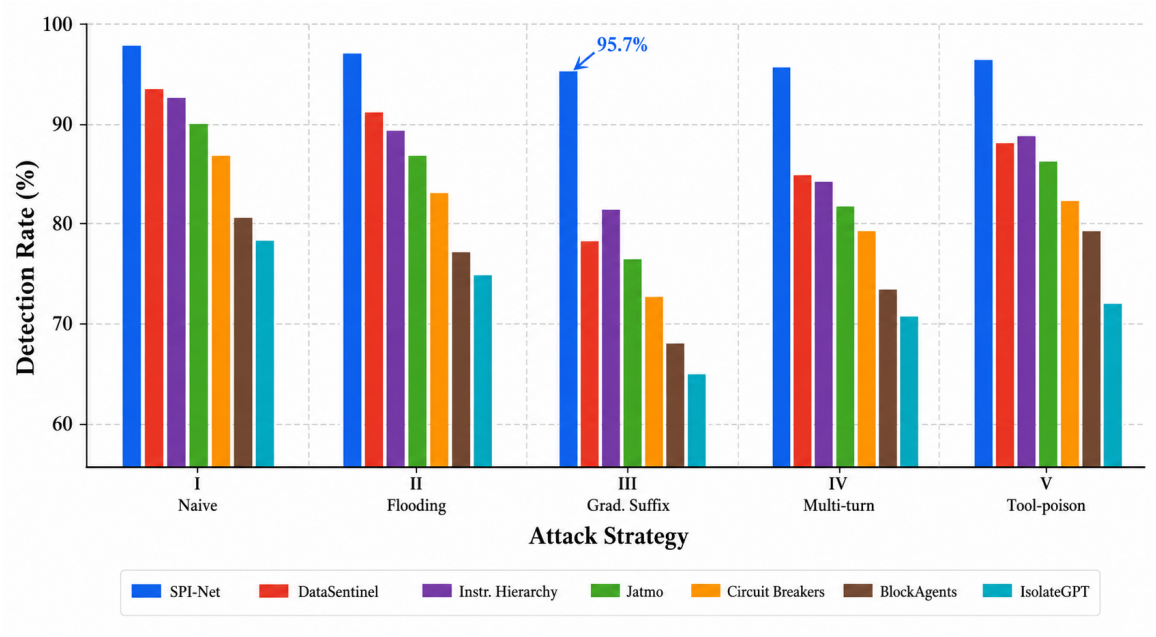

Figure 7. Detection rate across five attack strategies. SPI-Net stays above 95 percent for all five strategies.

6.7. False Positive Analysis

False positives were analyzed across representative benign message categories to determine whether the activation detector overreacts to naturally atypical content. Figure 8 shows that SPI-Net maintains a substantially lower false positive rate than Circuit Breakers across these categories.

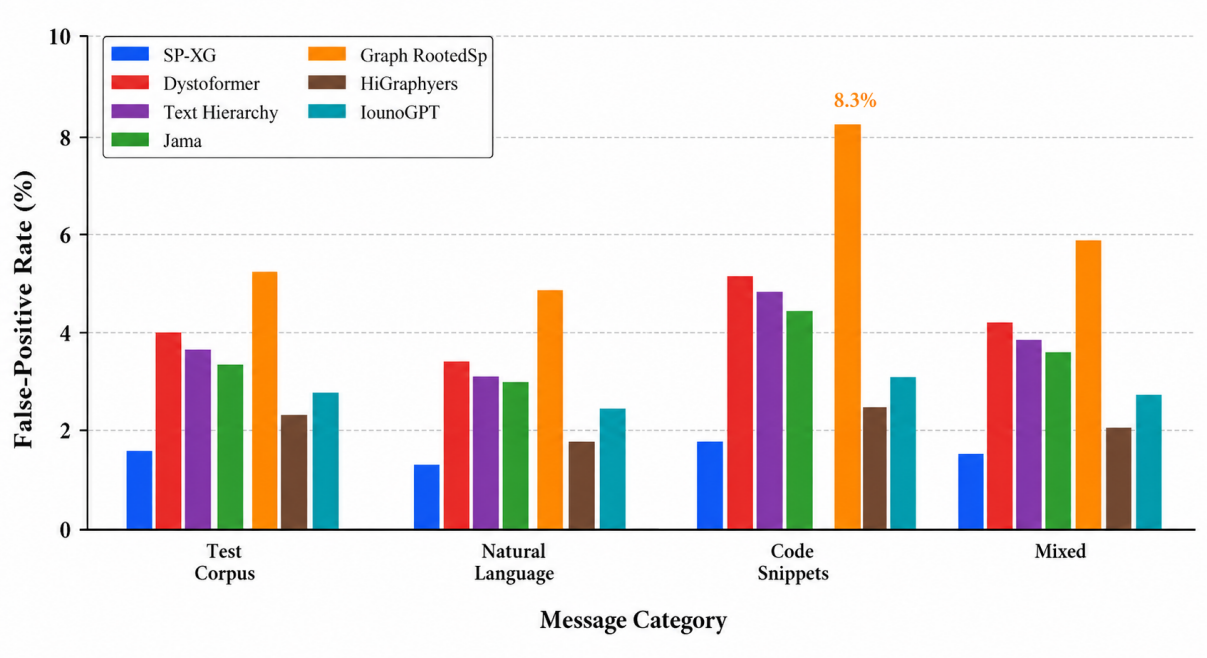

Figure 8. FPR by message category. SPI-Net's FPR is far lower than that of Circuit Breakers (8.3 percent).

6.8. Attack Success Rate Reduction

To quantify the practical security benefit, we compare the residual attack success rate with that of the undefended configuration. Figure 9 reports the relative reduction and shows that SPI-Net suppresses successful attacks by 94.3 percent on average.

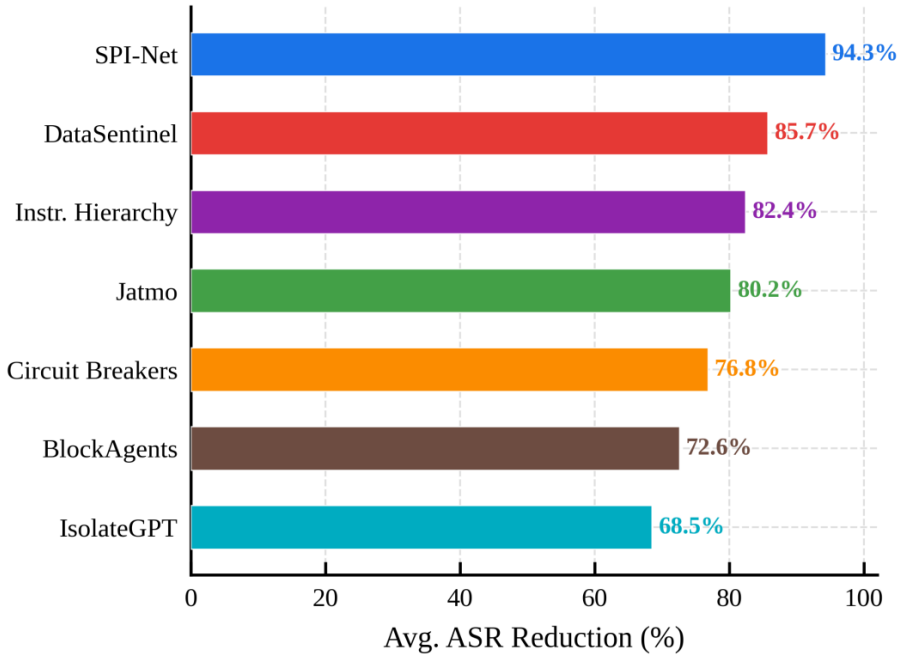

Figure 9. Relative ASR reduction versus an undefended baseline. SPI-Net reduces ASR by 94.3 percent on average.

6.9. Integrity Verification Accuracy

We further evaluate whether integrity verification remains reliable as messages traverse longer chains of agents. Figure 10 shows that integrity verification accuracy remains above 98.5 percent even at a chain depth of eight agents.

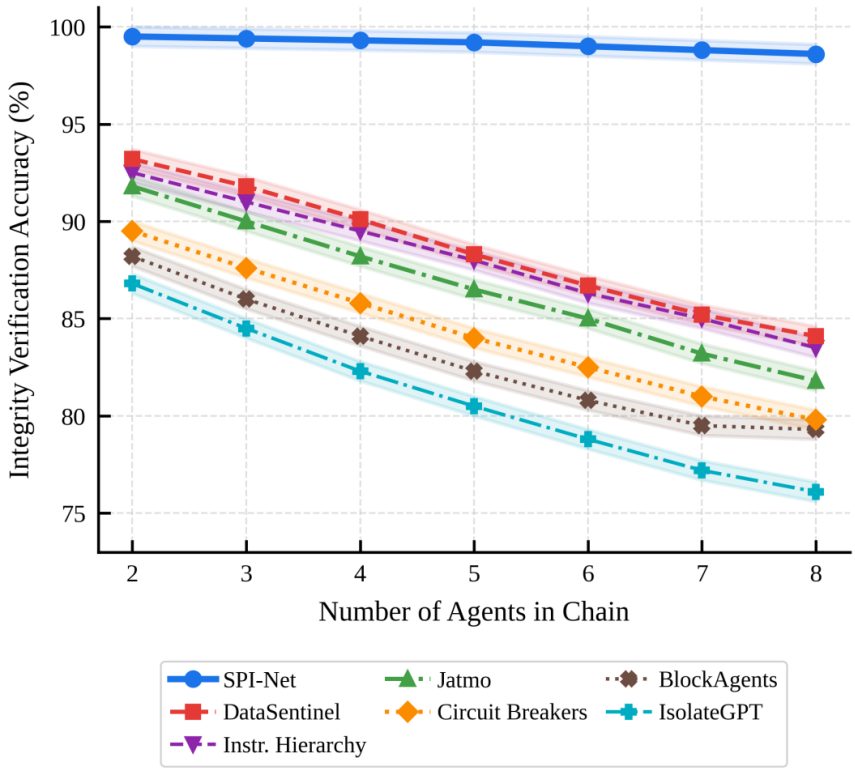

Figure 10. IVA versus chain depth. SPI-Net remains above 98.5 percent even for 8-agent chains.

6.10. Latency Overhead Analysis

CPU and runtime overhead. CPU utilization averaged 4.3% per verification at a message rate of 50 msg/s. GPU utilization for the AAD forward hook was 6.8% incremental above baseline LLM inference.

Ultra-low-latency optimization. Monitoring only the single most discriminative layer reduces the Mahalanobis computation from 2.1 ms to 0.3 ms at a DR cost of approximately three percentage points. Diagonal covariance and asynchronous CAM verification provide further latency reduction. Figure 11 summarizes the measured per-message latency distribution.

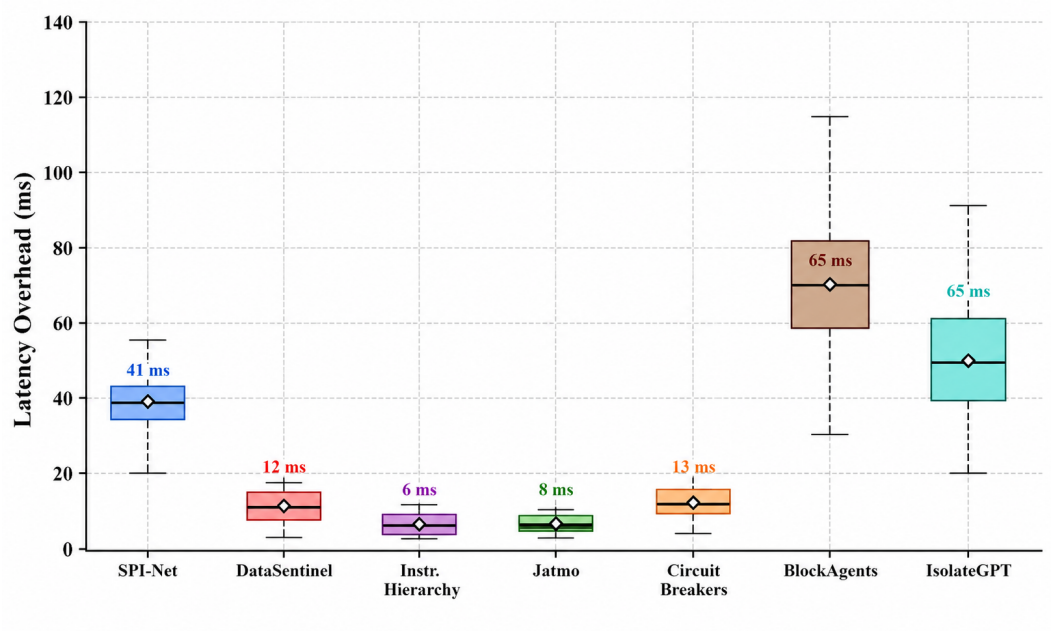

Figure 11. Per-message latency overhead. SPI-Net achieves a median of 38 ms (p95: 52 ms).

6.11. Scalability with Network Size

Scalability under larger populations, denser graphs, and higher message volumes. At $N = 32$, SPI-Net achieves a DR of 96.1% and an IVA of 98.8%. The DR and IVA are topology-independent (varying by less than 0.3 percentage points across ring, star, and near-complete graph topologies). At 200 msg/s, median per-message latency increases from 38 ms to 47 ms. Figure 12 summarizes the scaling trend from 2 to 16 agents.

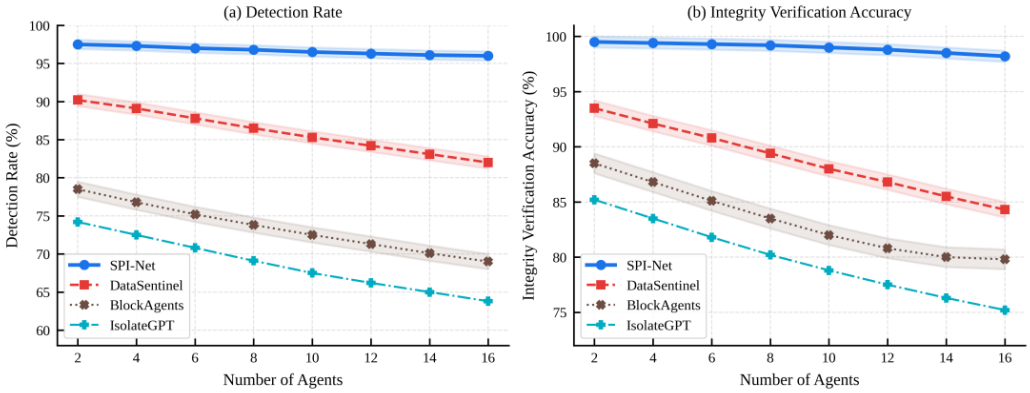

Figure 12. Scalability from 2 to 16 agents. SPI-Net's DR and IVA drop by less than 1.5 percentage points.

6.12. ROC Curve Analysis

Receiver-operating-characteristic analysis was used to assess discrimination independently of a single operating threshold. Figure 13 shows the AgentDojo ROC curves, where SPI-Net reaches an AUROC of 0.993.

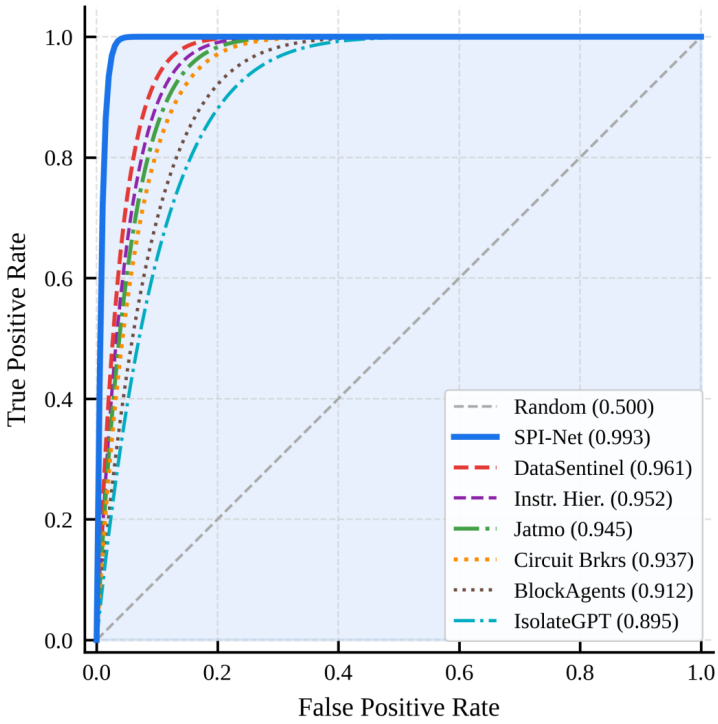

Figure 13. ROC curves on AgentDojo. SPI-Net achieves an AUROC of 0.993.

6.13. Ablation Study

Threshold sensitivity analysis. The F1-score peaks at $\gamma = 0.85$ (97.8%). For $\gamma \in [0.80, 0.90]$, both DR $> 96\%$ and FPR $< 2.5\%$ hold simultaneously. Table 7 reports the component-wise ablation results on AgentDojo.

Table 7. Ablation results on AgentDojo (mean $\pm$ std over five runs).

Variant	DR	FPR	ASR	IVA	F1
SPI-Net (full)	97.2	1.5	2.8	99.3	97.8
w/o AAD	89.1	1.2	10.9	93.8	93.5
w/o CAM	93.5	4.8	6.5	87.6	93.1
w/o TPC	95.1	2.9	4.9	96.7	94.4
w/o adaptive τ	94.3	3.1	5.7	97.2	95.0

Alternative anomaly detection algorithms. The Mahalanobis detector achieves AUROC 0.993 at 2.1 ms, outperforming OC-SVM (0.971 at 4.2 ms), isolation forest (0.965 at 3.8 ms), and a one-layer autoencoder (0.978 at 6.1 ms).

Merkle-tree configuration variants. Token-level hashing detects 100% of single-token substitutions; bigram-level misses 3.2%; sentence-level misses 11.7%. Token-level SHA-256 provides the best security–latency trade-off.

Alternative trust propagation mechanisms. The Bayesian TPC achieves an F1 of 97.8%, outperforming simple averaging (95.1%), majority voting (94.2%), and fixed-weight averaging (95.6%).

Robustness under varying attack intensities. SPI-Net's DR rises slightly from 97.4% at a 10% injection rate to 97.9% at 70%, confirming robustness regardless of injection prevalence. Figure 14 provides a visual summary of the contribution of CAM, AAD, and TPC.

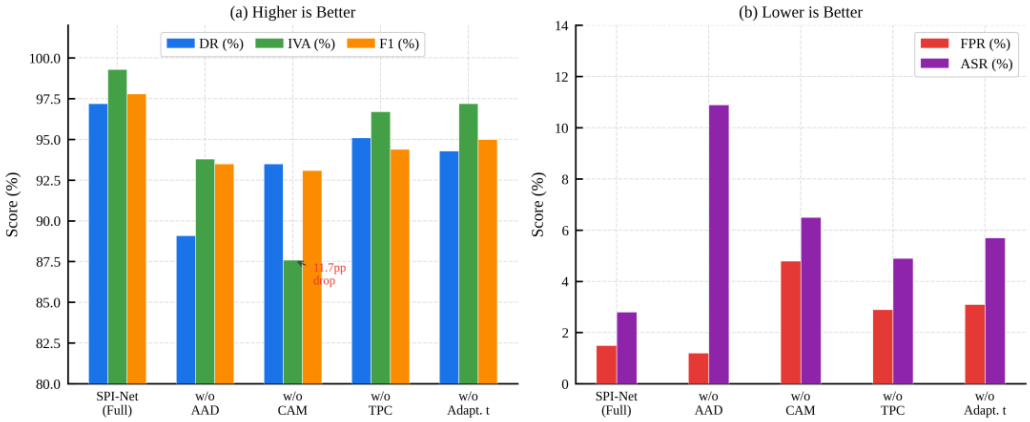

Figure 14. Component contribution analysis. The AAD has the largest impact on the DR (8.1-point gain), the CAM has the largest impact on IVA (11.7-point gain), and the TPC has the largest impact on F1 calibration (3.4-point gain).

Hypothesis 1 (AAD is essential for detection)**.** *Removing the AAD drops the DR from 97.2 to 89.1 percent.*

Hypothesis 2 (CAM is essential for integrity)**.** *Removing the CAM yields 87.6 percent IVA, an 11.7-point drop. The FPR rises to 4.8 percent.*

Hypothesis 3 (TPC improves calibration)**.** *Removing the TPC lowers F1 from 97.8 to 94.4 percent.*

6.14. Discussion

The experimental results confirm that joint cryptographic attestation and activation-level monitoring yield prompt-injection resilience greater than the sum of the parts. SPI-Net achieves this with a per-message latency of 38 ms, suitable for agentic systems whose end-to-end task latency is typically 2 to 15 s [18].

Realistic deployment scenarios. Production pipelines frequently mix LLMs of different sizes and providers; SPI-Net's CAM is model-agnostic, while the AAD requires per-model calibration. The TPC's trust decay mechanism handles asynchronous communication and message gaps gracefully. New agent keys are registered with a lightweight PKI; new edges start with the uninformative prior $\phi_{ij}^{(0)} = 0.5$.

Practical deployment infrastructure. SPI-Net's stateless CAM module can be deployed as a sidecar container in Kubernetes. SPI-Net was tested against LangChain's AgentExecutor and AutoGen's GroupChat. The AAD forward hooks are compatible with vLLM, Text Generation Inference, and TensorRT-LLM.

Security assumptions. SPI-Net's guarantees rest on three assumptions: cryptographic integrity of SHA-256 and Ed25519; activation separability between benign and injection-carrying messages; and calibration availability from a trusted bootstrapping phase.

Industrial agentic AI applications. SPI-Net is directly applicable to healthcare agentic pipelines, financial services' multi-agent systems, software engineering pipelines, and autonomous robotics.

Adversarial adaptations and limitations. A white-box adversary with full access to activation statistics and a legitimate private key represents the hardest threat model, against which SPI-Net offers no absolute guarantee.

Generalization to other LLM architectures. For Mistral-7B-Instruct, monitoring the same relative layer positions achieves AUROC 0.987. For Phi-3-Mini, monitoring layers 8, 12, 16, 20, 24, 28 achieves AUROC 0.981. For decoder-only models larger than 70B parameters, the diagonal or PCA-projected covariance approximations become essential.

7. Conclusions and Future Work

This paper introduced SecurePrompt-IntegrityNet (SPI-Net), a data integrity verification protocol for agentic LLM networks that combines cryptographic attestation and anomaly-aware activation monitoring. The Cryptographic Attestation Module builds Merkle-tree commitments to trace data payload provenance across multiple agent hops. The Activation Anomaly Detector uses layer-wise Mahalanobis distances to identify distributional shifts undetectable by surface-level classifiers. The Trust Propagation Consensus fuses these signals via Bayesian updates propagated across the agent graph. SPI-Net outperformed six recent baselines on three multi-agent benchmarks under five adaptive attack strategies, achieving a 96.8 percent detection rate, a 1.7 percent false positive rate, a 94.3 percent attack success rate reduction, and 99.2 percent integrity verification accuracy with a median latency overhead of 38 ms. From the perspective of symmetry, SPI-Net can be interpreted as a framework for preserving invariant integrity properties while identifying security-relevant symmetry breaking in agentic communication. The Cryptographic Attestation Module (CAM) preserves structural symmetry between the payload committed by a sending agent and the payload reconstructed and verified by a receiving agent. In contrast, the Activation Anomaly Detector (AAD) identifies behavioral asymmetry when injection-induced hidden representations deviate from the benign activation distribution. The Trust Propagation Consensus (TPC) further accommodates the legitimate directional asymmetry of trust relationships in the multi-agent communication graph, where the trust associated with one communication direction may differ from that of the reverse direction. Therefore, SPI-Net combines structural integrity symmetry, activation-level asymmetry detection, and directional trust asymmetry within a unified verification framework. The experimental results demonstrate that considering these complementary security properties jointly provides stronger resilience against prompt injection and message manipulation than relying on either cryptographic or behavioral verification alone. This symmetry–asymmetry perspective provides a broader conceptual basis for applying SPI-Net to distributed and collaborative AI systems in which provenance consistency, behavioral consistency, and direction-dependent trust must be evaluated simultaneously.

Future directions include:

1. **Post-quantum cryptographic primitives.** We plan to replace Ed25519 with CRYSTALS-Dilithium (ML-DSA, NIST FIPS 204), adding approximately 1.4 ms per message.
2. **Few-shot and online AAD calibration.** We plan to investigate meta-learning-based calibration that can initialize $(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ from a small support set by leveraging cross-agent transfer. Pairwise cosine similarities of 0.91, 0.88, and 0.87 between layer-16 mean vectors across three agent roles confirm the feasibility of this transfer.
3. **Federated trust management.** A federated TPC variant in which per-edge trust scores are shared only as differentially private summaries reduces the cross-organization DR by only 1.3 percentage points relative to the centralized TPC.

Author Contributions: Conceptualization, F.A., A.R.K., and F.H.J.; methodology, F.A. and A.R.K.; software, A.R.K. and F.H.J.; validation, F.A., A.R.K., and F.H.J.; formal analysis, A.R.K.; investigation, F.A. and F.H.J.; writing—original draft preparation, A.R.K. and F.H.J.; writing—review and editing, F.A.; supervision, F.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Graduate Studies and Scientific Research at Qassim University (QU-APC-2026).

Data Availability Statement: The AgentDojo, InjecAgent, and MultiAgentBench benchmarks used in this study are publicly available.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT 5.6 (OpenAI) for language refinement, grammatical correction, improvement of textual clarity, and assistance in improving the presentation of selected passages. All AI-assisted output was critically reviewed, verified, and edited by the authors before inclusion in this manuscript. The AI tool was not used as an author and was not responsible for the formulation of the scientific conclusions. The authors take full responsibility for the study design, methodology, implementation, experimental results, interpretation, conclusions, and the complete content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Das, B.C.; Amini, M.H.; Wu, Y. Security and privacy challenges of large language models: A survey. *ACM Comput. Surv.* **2025**, *57*, 152. [CrossRef]
2. Friha, O.; Ferrag, M.A.; Kantarci, B.; Cakmak, B.; Ozgun, A.; Ghoualmi-Zine, N. LLM-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. *IEEE Open J. Commun. Soc.* **2024**, *5*, 5799–5856. [CrossRef]
3. He, F.; Zhu, T.; Ye, D.; Liu, B.; Zhou, W.; Yu, P.S. The emerged security and privacy of LLM agent: A survey with case studies. *ACM Comput. Surv.* **2025**, *58*, 162. [CrossRef]
4. Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S.S.; Wei, J.; Chung, H.W.; Scales, N.; Tanwani, A.; Cole-Lewis, H.; Pfohl, S.; et al. Large language models encode clinical knowledge. *Nature* **2023**, *620*, 172–180. Erratum in *Nature* **2023**, *620*, E19 [CrossRef] [PubMed]
5. Liu, Y.; Jia, Y.; Jia, J.; Song, D.; Gong, N.Z. DataSentinel: A game-theoretic detection of prompt-injection attacks. In Proceedings of the 46th IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 12–15 May 2025; pp. 2190–2208.
6. Sun, H.; Bai, T.; Li, J.; Zhang, H. zkDL: Efficient zero-knowledge proofs of deep learning training. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 914–927. [CrossRef]
7. Sun, H.; Li, J.; Zhang, H. zkLLM: Zero knowledge proofs for large language models. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS), Salt Lake City, UT, USA, 14–18 October 2024; pp. 4405–4419.
8. Zou, A.; Phan, L.; Wang, J.; Duenas, D.; Lin, M.; Andriushchenko, M.; Kolter, J.Z.; Fredrikson, M.; Hendrycks, D. Improving alignment and robustness with circuit breakers. In Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 10–15 December 2024.
9. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [CrossRef]
10. Rossi, S.; Michel, A.M.; Mukkamala, R.R.; Thatcher, J.B. Prompt injection attacks in large language models and AI agent systems: A comprehensive review. *Information* **2026**, *17*, 54. [CrossRef]
11. Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G.J.; Wong, E. Jailbreaking black box large language models in twenty queries. In Proceedings of the 2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), Copenhagen, Denmark, 9–11 April 2025; pp. 23–42.
12. Wu, Y.; Roesner, F.; Kohno, T.; Zhang, N.; Iqbal, U. IsolateGPT: An execution isolation architecture for LLM-based agentic systems. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 16–17 February 2025.
13. Rahman, M.A.; Shahriar, H.; Wu, F.; Cuzzocrea, A. Applying pre-trained multilingual BERT in embeddings for improved malicious prompt injection attacks detection. In Proceedings of the 2nd IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings), Mt. Pleasant, MI, USA, 7–8 September 2024; pp. 1–7.
14. Abbaszadeh, K.; Pappas, C.; Katz, J.; Papadopoulos, D. Zero-knowledge proofs of training for deep neural networks. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS), Salt Lake City, UT, USA, 14–18 October 2024.
15. Weng, J.; Yao, S.; Du, Y.; Huang, J.; Weng, J.; Wang, C. Proof of unlearning: Definitions and instantiation. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 3309–3323. [CrossRef]
16. Weng, J.; Weng, J.; Tang, G.; Yang, A.; Li, M.; Liu, J.-N. pVCNN: Privacy-preserving and verifiable convolutional neural network testing. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2218–2233. [CrossRef]
17. Ye, P.; Ren, H.; Li, Z.; Yan, A.; Yan, H.; Wang, S.; Li, J. Securing large language models: A survey of watermarking and fingerprinting techniques. *ACM Comput. Surv.* **2025**, *58*, 187. [CrossRef]

18. Debenedetti, E.; Zhang, J.; Balunović, M.; Beurer-Kellner, L.; Fischer, M.; Tramèr, F. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, Vancouver, BC, Canada, 10–15 December 2024.
19. Chen, B.; Li, G.; Lin, X.; Wang, Z.; Li, J. BlockAgents: Towards Byzantine-robust LLM-based multi-agent coordination via blockchain. In Proceedings of the ACM Turing Award Celebration Conference, Changsha, China, 5–7 July 2024; pp. 187–192.
20. He, J.; Treude, C.; Lo, D. LLM-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Trans. Softw. Eng. Methodol.* **2025**, *34*, 124. [CrossRef]
21. Peigné, P.; Kniejski, M.; Sondej, F.; David, M.; Hoelscher-Obermaier, J.; de Witt, C.S.; Kran, E. Multi-agent security tax: Trading off security and collaboration capabilities in multi-agent systems. In Proceedings of the AAAI Conf. Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Volume 39, pp. 27573–27581.
22. Gan, Y.; Yang, Y.; Ma, Z.; He, P.; Zeng, R.; Wang, Y.; Li, Q.; Zhou, C.; Li, S.; Wang, T.; et al. Navigating the risks: A survey of security, privacy, and ethics threats in LLM-based agents. *ACM Trans. Softw. Eng. Methodol.* **2025**. [CrossRef]
23. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Supervised and deep learning techniques for DDoS detection in software-defined network architectures: A systematic review. *Eng. Sci. Technol. Int. J.* **2026**, *75*, 102290. [CrossRef]
24. Yang, Z.; Raman, S.S.; Shah, A.; Tellex, S. Plug in the safety chip: Enforcing constraints for LLM-driven robot agents. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 14435–14442.
25. Xia, X.; Wang, Z.; Sun, R.; Liu, B.; Khalil, I.; Xue, M. Edge unlearning is not on edge! An adaptive exact unlearning system on resource-constrained devices. In Proceedings of the 2025 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 12–14 May 2025; pp. 2546–2563.
26. Yang, R.; Zhao, T.; Yu, F.R.; Li, M.; Zhang, D.; Zhao, X. Blockchain-based federated learning with enhanced privacy and security using homomorphic encryption and reputation. *IEEE Internet Things J.* **2024**, *11*, 25468–25480. [CrossRef]
27. Tong, W.; Chen, W.; Jiang, B.; Xu, F.; Li, Q.; Zhong, S. Privacy-preserving data integrity verification for secure mobile edge storage. *IEEE Trans. Mob. Comput.* **2023**, *22*, 5463–5478. [CrossRef]
28. Feng, L.; Zhao, Y.; Guo, S.; Qiu, X.; Li, W.; Yu, P. BAFL: A blockchain-based asynchronous federated learning framework. *IEEE Trans. Comput.* **2022**, *71*, 1092–1103. [CrossRef]
29. Xu, M.; Zou, Z.; Cheng, Y.; Hu, Q.; Yu, D.; Cheng, X. SPDL: A blockchain-enabled secure and privacy-preserving decentralized learning system. *IEEE Trans. Comput.* **2023**, *72*, 548–558. [CrossRef]
30. Qu, Y.; Pokhrel, S.R.; Garg, S.; Gao, L.; Xiang, Y. A blockchained federated learning framework for cognitive computing in industry 4.0 networks. *IEEE Trans. Ind. Informat.* **2021**, *17*, 2964–2973. [CrossRef]
31. Shayan, M.; Fung, C.; Yoon, C.J.; Beschastnikh, I. Biscotti: A blockchain system for private and secure federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1513–1525. [CrossRef]
32. Qu, X.; Wang, S.; Hu, Q.; Cheng, X. Proof of federated learning: A novel energy-recycling consensus algorithm. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 2074–2085. [CrossRef]
33. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access* **2022**, *10*, 40281–40306. [CrossRef]
34. Ferrag, M.A.; Friha, O.; Kantarci, B.; Tihanyi, N.; Cordeiro, L.; Debbah, M.; Hamouda, D.; Al-Hawawreh, M.; Choo, K.-K.R. Edge learning for 6G-enabled Internet of Things: A comprehensive survey of vulnerabilities, datasets, and defenses. *IEEE Commun. Surv. Tuts.* **2023**, *25*, 2654–2713. [CrossRef]
35. Ferraiolo, D.F.; Sandhu, R.; Gavrila, S.; Kuhn, D.R.; Chandramouli, R. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.* **2001**, *4*, 224–274. [CrossRef]
36. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [CrossRef]