The Adversarial Trust Layer

Why the MCP Ecosystem Needs Cryptographic Attestation and Multi-Agent Verification

---

Position Paper — February 2026

Phil Stafford, AI Security Researcher

Singularity Systems

Abstract

The Model Context Protocol (MCP) is becoming the universal interface between AI agents and external tools — but it has no mechanism for verifying tool identity, provenance, or behavioral claims. Real supply chain attacks, critical CVEs, and cross-tenant data breaches have already been documented. This paper argues that the ecosystem needs two complementary layers: a public cryptographic trust registry (Credence) for static verification of MCP tool identity and attestation, and multi-agent adversarial debate (ThinkTank) for adversarial analysis that stress-tests scanner findings before a verdict is issued. Neither layer alone is sufficient. Together, they form a defense-in-depth architecture for the agentic AI era.

https://pestafford.github.io/credence-registry/

The Problem in One Sentence

The Model Context Protocol is becoming the nervous system of enterprise AI, and it has no immune system.

1. MCP Won — Now It Has to Survive

On February 5, 2026, security researchers at Straiker STAR Labs disclosed a supply chain attack that distilled every open problem in MCP security into a single campaign. A threat actor group known as SmartLoader — previously documented distributing info-stealers through deceptive installers — cloned a legitimate Oura Ring MCP server, built a network of at least five fake GitHub accounts with fabricated fork and contributor histories, and submitted the trojanized server to legitimate MCP registries including MCP Market. The attack was patient and methodical: three months building fake credibility before deploying the payload. The trojanized server delivered the StealC infostealer, harvesting browser passwords, SSH keys, API credentials, cryptocurrency wallets, and the health data of anyone who connected their Oura Ring through the compromised integration.

The details are instructive. The original Oura MCP server was built by an OpenAI engineer to let AI assistants query sleep and readiness data — exactly the kind of health-optimization tool that productivity-focused developers adopt without a second thought. SmartLoader's clone was functionally identical. Its source code matched the legitimate version. Its contributor list showed active community involvement. The only visible anomaly was what was *missing*: the original author wasn't listed as a contributor. A cryptographic provenance chain would have caught this instantly. Instead, the trojanized server sat in a public registry alongside legitimate tools, indistinguishable to any developer or agent evaluating it.

This wasn't the first warning. When Anthropic released the Model Context Protocol in November 2024, the pitch was simple: a universal standard for connecting LLMs to external tools. Think of it as USB-C for AI applications. The industry agreed. Microsoft integrated MCP across Copilot Studio and Azure AI Foundry. GitHub joined the MCP Steering Committee. Thousands of community-built MCP servers appeared within months, connecting AI agents to databases, file systems, APIs, cloud infrastructure, and each other.

The adoption curve worked. The security story didn't.

By mid-2025, the attack surface was wide open. Backslash Security reported hundreds of MCP servers misconfigured and exposed to the public internet, including the "NeighborJack" vulnerability where servers bound to 0.0.0.0 were accessible to any adjacent network device. Palo Alto Networks' Unit 42 team demonstrated three critical attack vectors through MCP's sampling feature alone: resource theft, conversation hijacking, and covert tool invocation — all achievable through a single malicious MCP server that looked identical to a legitimate one. JFrog disclosed CVE-2025-6514, a CVSS 9.6 command-injection vulnerability in mcp-remote — affecting over 437,000 downloads and every major MCP client including Claude Desktop, Cursor, and VS Code. The Practical DevSecOps team's MCPTox benchmark revealed that tool poisoning attacks — where malicious logic is embedded in a tool while preserving its legitimate interface — are alarmingly common across the ecosystem.

These aren't theoretical risks. AuthZed compiled the first consolidated timeline of MCP security breaches, documenting real incidents: a GitHub MCP integration that exfiltrated private repository contents, salary information, and internal project details through a public pull request. An Asana MCP bug that exposed one organization's data to another organization's users. Anthropic's own MCP Inspector tool was found to allow unauthenticated remote code execution (CVE-2025-49596, CVSS 9.4). And now SmartLoader has demonstrated that traditional supply chain attack playbooks — fake accounts, manufactured social proof, registry poisoning — transfer directly to the MCP ecosystem with no modifications required.

The pattern is consistent across every analysis: **MCP tools have no verifiable identity, no cryptographic proof of provenance, no attestation of behavioral claims, and no trust registry for cross-organizational discovery.** The protocol that's becoming the standard interface between AI agents and the world has no mechanism for an agent — or the human deploying it — to verify that a tool is what it says it is. As Straiker's own researchers concluded: the MCP ecosystem lacks the security infrastructure that has developed around traditional package managers — there is no equivalent to npm audit, Dependabot, or Snyk for MCP servers.

2. The Agent Identity Crisis

The MCP security gap exists within a broader crisis. AI agents are proliferating faster than the infrastructure to govern them.

Gartner projects that by the end of 2026, 40% of enterprise applications will be integrated with task-specific AI agents. AWS reports that more than half of enterprises now classify agentic AI as a strategic priority. According to Orca Security's analysis, non-human identities already outnumber human identities 50:1 in the average enterprise environment. CyberArk's 2025 Identity Security Landscape report measured the ratio at 82:1 — a figure that was projected as a future concern just months earlier but is already the documented reality.

The consequences of this ungoverned growth are already visible. A SailPoint survey of 353 enterprise security professionals found that 80% of organizations reported their AI agents taking unintended actions or behaving unexpectedly. In one incident described by researcher Akshay Mittal at MLOps World 2025, a single compromised agent in a 50-agent machine learning operations system triggered a complete cascade failure within six minutes — the rogue agent impersonated a model deployment service and pushed corrupted models downstream before monitoring agents could distinguish malicious traffic from legitimate operations.

The SmartLoader campaign against the Oura Ring MCP server demonstrates how these identity failures compound: a supply chain attack exploiting fabricated trust signals in one registry can deliver credential-stealing malware into the development environments of every organization that adopts the compromised tool. The npm ecosystem learned this lesson with the Shai-Hulud worm campaign targeting AI development tooling; the MCP ecosystem is learning it now.

The identity and access management frameworks that enterprises rely on — OAuth 2.0, OpenID Connect, SAML — were designed for a world of deterministic applications and single authenticated principals. As ISACA's analysis notes, agentic AI violates those assumptions in fundamental ways:

agents have dynamic identity lifecycles (spawning and retiring in seconds), require capability attestation (not just identity verification), and operate across diverse protocols including MCP, A2A, and ACP simultaneously.

The industry recognizes the gap. The IETF has a draft standard for an Agent Name Service (ANS). GoDaddy built a proof-of-concept ANS Registry with a Registration Authority and transparency logs. HID Global's PKI Market Study found that AI agent certificates represent a stronger enterprise trend than even post-quantum cryptography — 15% of organizations have already begun deploying certificates for agents. Indicio, an NVIDIA Inception partner, is building ProvenAI with Verifiable Credentials for agent-to-agent authentication. Prove launched Verified Agent for the agentic commerce market, with Mastercard publicly stating that "payments must be native to the agentic experience."

NIST has issued a Request for Information on AI agent security assessment methods, due March 2026. OWASP published its Top 10 for Agentic Applications 2026, listing memory poisoning, tool misuse, and privilege compromise as primary risks. The OpenID Foundation is pushing AI-specific IAM standards.

Everyone agrees the problem is real. No one has shipped the MCP-specific trust layer yet.

3. The Case for a Public Cryptographic Trust Registry

The existing supply chain security stack — Sigstore for cryptographic signing, SLSA for build provenance, in-toto for pipeline attestation — was designed for traditional software artifacts: container images, packages, binaries. The Coalition for Secure AI (CoSAI), whose members include Google, Microsoft, NVIDIA, and Cisco, has extended this thinking to AI model signing, defining three verification aspects: integrity (the model hasn't been modified), provenance (traceable development lifecycle), and properties (verifiable claims about performance and compliance). Google's own AI supply chain guidance calls tamper-proof provenance "indispensable." Red Hat's Trusted Artifact Signer 1.3 now extends cryptographic signing to ML models and inference frameworks.

But none of this reaches the MCP tool layer. A signed model deployed behind an unsigned, unattested MCP server is a locked door in an open field.

What the ecosystem needs is a **public** cryptographic trust registry for MCP — not a gated commercial product, but shared infrastructure that raises the security floor for every participant, the same way Let's Encrypt made TLS certificates accessible to the entire web and Sigstore made software signing practical for every open-source maintainer. The security of the MCP ecosystem cannot depend on whether individual organizations can afford to buy trust verification. It has to be foundational.

This is the design philosophy behind **Credence**, an open, public trust registry for MCP tools built on three capabilities:

Tool Identity Attestation. Every MCP server registered with Credence receives a cryptographic identity that binds its tool definitions, capabilities, and behavioral claims to a verifiable provenance chain. When an AI agent discovers an MCP tool through Credence, it can verify that the tool was published by a known entity, that its current definition matches what was attested, and that it hasn't been modified since attestation — directly addressing the "rug pull" risk documented by Backslash Security. Credence's MCP tool analyzer detects precursor patterns for rug pulls — dynamic description

loading, version-gated tool registration, time-based conditionals — and hashes every tool definition at attestation time so that changes between scans are detectable.

Behavioral Trust Scoring. Beyond static identity, Credence computes a three-dimensional trust score — security, provenance, and behavioral — based on static analysis of the server's tool definitions, dependency tree, and source code patterns. The behavioral dimension specifically evaluates MCP tool definitions for suspicious patterns: dynamic description loading, version-gated tool registration, prompt injection signatures, and schema manipulation. These scores are cryptographically signed and published alongside the attestation, giving agents and developers quantitative trust data instead of social signals. This addresses the fundamental problem identified in the arXiv MCP landscape survey: users currently delegate tool selection entirely to AI applications without cryptographic verification or contextual awareness.

Open Trust Registry for Discovery. Credence functions as a publicly accessible, trust-aware discovery layer for MCP tools, analogous to what ANS proposes for general agent discovery but specific to the MCP protocol's tool ecosystem. Any developer, organization, or agent can query the registry to evaluate the trust posture of an MCP server before connecting to it. Organizations can implement automated policies around Credence verdicts — the CLI provides structured exit codes (0=verified, 1=unattested, 2=flagged, 3=rejected) for CI pipeline integration, and the MCP server interface lets agents query trust posture programmatically before connecting. This is the risk mitigation pattern that CoSAI recommends but that no MCP-specific infrastructure has delivered.

The public nature of this registry is a deliberate architectural decision. MCP's security is a commons problem. A compromised MCP server doesn't just affect the organization that deployed it — it affects every agent that connects to it, every user whose data flows through it, and every downstream system that trusts its outputs. Private, per-customer trust verification creates islands of safety surrounded by an ocean of unverified tools. A public registry creates herd immunity.

4. The Limits of Static Trust (and Why Dynamic Verification Matters)

Cryptographic attestation is necessary. It is not sufficient.

Consider a scenario: an MCP server is published by a known vendor, cryptographically signed, and registered with valid attestations. Its tool definitions pass all static verification checks. But its descriptions contain subtle prompt injection patterns that, when processed by an LLM, cause the agent to exfiltrate context window contents to an external endpoint. The attestation is technically accurate — the tool does what it claims — but the *way* it does it embeds a security risk that static analysis alone cannot reliably detect.

This is where the research on multi-agent debate becomes directly relevant to security.

The foundational work by Du et al., published at ICML 2024, demonstrated that multiple LLM instances debating their individual responses over multiple rounds significantly reduces hallucination and improves factual accuracy compared to any single model's output. Subsequent research has refined this finding considerably. Zhou and Chen's Adaptive Heterogeneous Multi-Agent Debate framework,

published in November 2025, showed that assigning diverse, specialized roles to debate agents — rather than using homogeneous agents with majority voting — achieves 4–6% higher accuracy and over 30% fewer factual errors. Hegazy (2024) and Zhang et al. (2025) demonstrated that deploying agents based on different foundation models yields 91% accuracy versus 82% with homogeneous agents on mathematical reasoning tasks, with emergent teacher-student dynamics between models.

Zhang et al.'s systematic evaluation of five MAD frameworks across nine benchmarks, presented at ICLR 2025, added important nuance: multi-agent debate's advantage is strongest on harder tasks and in safety-critical domains where the cost of a wrong answer is highest. Moderate, calibrated disagreement outperforms both maximal adversarial opposition and cooperative agreement-seeking. Hu et al.'s adaptive stability detection framework, accepted at NeurIPS 2025, formally proved that debate amplifies correctness over static ensembles under mild assumptions, providing mathematical grounding for what practitioners observe empirically. Wu et al. (2025) further confirmed through controlled experiments that intrinsic reasoning strength and group diversity are the dominant drivers of debate success.

These findings have direct application to security analysis. **ThinkTank** is a multi-agent debate system that applies structured adversarial reasoning to security questions. Five agents — each with a distinct analytical role and perspective bias — debate the findings from Credence's static analysis pipeline across up to five rounds, with early termination when positions stabilize.

Two skeptic agents (an Adversarial Attacker primed with real-world MCP attack patterns and a Supply Chain Analyst focused on provenance anomalies) construct the strongest case for risk. Two believer agents (a Devil's Advocate and a Pattern Matcher) construct the strongest case that the findings are benign. A neutral Compliance Reviewer evaluates both sides, identifies gaps in reasoning, and synthesizes a confidence-scored verdict with dissenting opinions preserved. The role heterogeneity is deliberate: Zhou and Chen's A-HMAD research demonstrates that assigning diverse, specialized roles to debate agents — rather than using homogeneous agents — achieves 4–6% higher accuracy and over 30% fewer factual errors. The architecture is designed so that cross-model heterogeneity can be layered on as additional foundation models become viable for security analysis.

This isn't academic novelty seeking a problem. The offensive side of this equation is already here. Security Boulevard documented in December 2025 that nation-state actors and sophisticated cybercriminals now orchestrate five to eight different LLMs simultaneously in adaptive breach campaigns — different models handling social engineering, malware creation, OSINT gathering, and credential operations. Defenders using a single model instance for security analysis face an asymmetric disadvantage against coordinated multi-model attack chains.

The multi-agent approach is validated by security-specific research as well. A ScienceDirect study (March 2025) demonstrated that a cross-LLM architecture achieves 98% accuracy in security assessment tasks by leveraging complementary analytical perspectives. The ZeroDay-LLM framework (MDPI, October 2025) showed that ensemble defense mechanisms achieve 97.8% detection accuracy with 23% fewer false positives than single-instance approaches. Role diversity — which ThinkTank implements today — captures a significant portion of this advantage; cross-model diversity, which the architecture supports as an upgrade path, would capture the rest.

5. The Adversarial Trust Layer: Static + Dynamic Verification

The combined value proposition of these two approaches is straightforward:

A public trust registry tells you WHO built an MCP tool and cryptographically verifies WHAT it claims to do. Multi-agent adversarial debate stress-tests the scanner findings and tells you whether to TRUST it.

Cryptographic attestation provides the static trust layer — proof of identity, provenance, and trust scoring. Structured adversarial debate provides the analytical layer — multi-agent reasoning that stress-tests scanner findings, identifies inconsistencies between signals, and surfaces risks that static rules alone cannot catch.

No existing solution combines these capabilities:

- ANS (IETF draft) provides agent discovery, but not MCP-specific tool attestation.
- Sigstore and SLSA provide build provenance, but not adversarial analysis of security findings.
- MCPTox provides tool poisoning benchmarks, but not continuous adversarial analysis.
- Palo Alto publishes MCP vulnerability research, but doesn't operate a trust registry.
- GoDaddy built an ANS proof-of-concept, but without integrated adversarial analysis.

The combination creates a trust pipeline that addresses the full threat model: static verification catches known-bad actors and tampered artifacts; dynamic adversarial analysis catches sophisticated attacks that pass static checks. This defense-in-depth approach mirrors what the broader security industry has learned over decades — no single layer is sufficient, and the interaction between layers catches what individual layers miss.

When a public attestation registry feeds into adversarial analysis, the system can answer questions that neither approach answers alone: "This MCP server is cryptographically signed by a known vendor, but do the scanner findings hold up under adversarial scrutiny? Do the tool descriptions contain patterns consistent with prompt injection? Does the declared capability scope match the actual API surface?" These are the questions that the SmartLoader supply chain attack, the GitHub MCP exfiltration, and the Asana cross-tenant breach would have forced someone to ask — if anyone had the tooling to ask them.

6. Where This Stands

The regulatory and standards environment is accelerating. NIST's RFI on AI agent security is due March 2026. OWASP's Top 10 for Agentic Applications is published. The OpenID Foundation is pushing AI-specific IAM standards. The EU's proposed GDPR amendments address AI agent data processing. CoSAI is standardizing model attestation. SLSA is expanding its track coverage. The convergence is clear: cryptographic trust infrastructure for AI is moving from recommendation to requirement.

The MCP ecosystem is standardizing but the security architecture is still being defined. Microsoft and GitHub's presence on the MCP Steering Committee signals long-term commitment to the protocol, but

the trust layer remains an open problem. The window to establish public trust infrastructure — before the ecosystem either ossifies without it or fragments around proprietary alternatives — is measured in months, not years.

The threat environment is not waiting. Documented incidents — from the SmartLoader supply chain attack on the Oura Ring MCP server to the GitHub MCP exfiltration and the CVSS 9.6 mcp-remote vulnerability — demonstrate that MCP-specific exploits, agent-to-agent compromise, and multi-model attack chains are operational realities, not future concerns. The $1.7 trillion agentic commerce market that payment networks are targeting cannot scale on the current trust infrastructure. The 80% of organizations reporting unexpected agent behavior are seeing the consequences of deploying autonomous systems without adequate verification.

The question isn't whether this infrastructure is needed. It's whether it arrives as public goods — accessible to every developer and organization participating in the MCP ecosystem — or as fragmented, proprietary solutions that protect paying customers while leaving the broader ecosystem exposed. The history of internet security, from the TLS certificate ecosystem to the open-source supply chain, consistently demonstrates that shared, public trust infrastructure outperforms private alternatives at protecting the commons.

The MCP ecosystem deserves the same treatment.

References

1. AuthZed, "A Timeline of Model Context Protocol Security Breaches," 2025
2. Backslash Security, "MCP Server Security: NeighborJack and Critical Misconfigurations," June 2025
3. Palo Alto Networks Unit 42, "Prompt Injection Attack Vectors Through MCP Sampling," December 2025
4. Palo Alto Networks, "Simplified Guide to MCP Vulnerabilities," October 2025
5. Red Hat, "Understanding MCP Security Risks and Controls," July 2025
6. Practical DevSecOps, "MCP Security Vulnerabilities," January 2026
7. Pillar Security, "The Security Risks of MCP," 2025
8. MCP Specification, "Security Best Practices (Draft)," modelcontextprotocol.io, 2025
9. Hou et al., "MCP: Landscape, Security Threats, and Future Research Directions," arXiv:2503.23278, October 2025
10. CoSAI, "Building Trust in AI Supply Chains: Model Signing," September 2025
11. Google Cloud, "Same Same but Also Different: Google Guidance on AI Supply Chain Security," October 2025
12. Trend Micro, "Exploiting Trust in Open-Source AI: The Hidden Supply Chain Risk No One Is Watching," August 2025
13. SLSA Framework, "Supply-chain Levels for Software Artifacts," slsa.dev, 2025
14. Red Hat, "Trusted Artifact Signer 1.3," November 2025
15. InfoQ, "Supply Chain Security: Provenance Tools Becoming Standard in Developer Platforms," August 2025
16. IDC, "Confidential Computing & Cryptographic Attestation," IDC #US53866125, November 2025
17. HID Global, "Trust Standards Evolve: AI Agents, Next Chapter for PKI," November 2025
18. ISACA, "The Looming Authorization Crisis: Why Traditional IAM Fails Agentic AI," December 2025
19. Strata, "Agentic AI Security: 8 Strategies in 2026," January 2026
20. WebProNews, "AI Agents' Trust Reckoning: One Hack Fells 50," January 2026
21. GoDaddy, "Building Trust at Internet Scale: Agent Name Service Registry," October 2025
22. Indicio, "Why Verifiable Credentials Will Power Real-World AI in 2026," January 2026
23. Solutions Review, "Identity Security Predictions for 2026," January 2026
24. Prove/Financial IT, "Verified Agent for Agentic Commerce," October 2025
25. Du et al., "Improving Factuality and Reasoning through Multiagent Debate," ICML 2024
26. Zhou & Chen, "Adaptive Heterogeneous Multi-Agent Debate (A-HMAD)," Journal of King Saud University, November 2025
27. Hegazy, "Diversity of Thought Elicits Stronger Reasoning Capabilities in Multi-Agent Debate Frameworks," IJCSMA, November 2024
28. Zhang et al., "Stop Overvaluing Multi-Agent Debate — We Must Rethink Evaluation and Embrace Model Heterogeneity," arXiv:2502.08788, February 2025; also presented as ICLR 2025 Blogpost
29. Hu et al., "Multi-Agent Debate for LLM Judges with Adaptive Stability Detection," NeurIPS 2025
30. Wu et al., "Can LLM Agents Really Debate? A Controlled Study of Multi-Agent Debate in Logical Reasoning," November 2025
31. Almufareh et al., "A Novel System for Strengthening Security in LLMs Against Hallucination and Injection Attacks," ScienceDirect, March 2025
32. Security Boulevard / Seceon, "Fighting AI with AI: The Rise of Multi-LLM Orchestrated Cyber Attacks," December 2025
33. Alsuwaiket, "ZeroDay-LLM: A Large Language Model Framework for Zero-Day Threat Detection in Cybersecurity," MDPI Information 16(11), October 2025
34. Hong & Oh, "Optimization for Threat Classification of Various Data Types-Based on ML Model and LLM," Nature Scientific Reports 15:22768, July 2025
35. WJARR, "Review of Generative AI for Multimodal Cybersecurity Threat Simulation," World Journal of Advanced Research and Reviews 27(01), July 2025
36. Straiker STAR Labs, "SmartLoader Clones Oura Ring MCP to Deploy Supply Chain Attack," February 5, 2026

37. CyberArk, "2025 Identity Security Landscape," April 2025
38. JFrog, "CVE-2025-6514: Command Injection in mcp-remote," July 2025
39. Tenable/Oligo Security, "CVE-2025-49596: MCP Inspector Unauthenticated RCE," 2025