Opaque Trust Inference for Autonomous AI Agents

A Formal Framework

AgentAnchor Research

Version 1.0 | January 2025

Preprint

Abstract

As autonomous AI agents increasingly operate in high-stakes environments—executing financial transactions, managing infrastructure, and making consequential decisions—the need for reliable trust measurement becomes critical. Yet existing approaches require access to agent internals, creating an "opacity barrier" that precludes trust assessment of proprietary, API-accessed, or black-box systems. We present the **Agent Trust Scoring Framework (ATSF)**, a methodology for inferring trust from observable behavior alone, without requiring access to model weights, training data, or decision-making internals.

ATSF introduces a four-tier observation model that maps deployment architectures to maximum achievable trust ceilings, a five-level trust progression system with formally specified advancement criteria, and novel algorithms for detecting behavioral degradation, Sybil attacks, and deceptive agent behavior. We provide a complete TLA+ formal specification and verify six safety invariants across 303,819 distinct system states. Property-based testing with 2,050+ generated test cases confirms all specified properties hold. Our jailbreak vulnerability assessment framework, comprising 21 probes across 10 attack categories, enables systematic security evaluation of LLM-based agents.

This work establishes theoretical foundations for third-party AI agent certification and provides a practical pathway toward insurable, auditable autonomous systems.

Keywords: AI trust, autonomous agents, formal verification, behavioral observation, trust scoring, agent certification

Contents

1. Introduction

1.1 The Trust Problem in Autonomous AI

The deployment of autonomous AI agents presents a fundamental trust problem: how can principals (operators, users, regulators, insurers) assess the trustworthiness of an agent whose decision-making process is opaque? This challenge is particularly acute for:

- **Proprietary API agents** where model weights and training data are inaccessible
- **Fine-tuned systems** where modifications may introduce unpredictable behaviors
- **Multi-agent compositions** where trust must propagate through delegation chains
- **Long-running autonomous systems** where behavioral drift may occur over time

Traditional software assurance approaches—code review, static analysis, formal verification of source—assume transparency into system internals. These methods fail when applied to agents built on large language models (LLMs) or other opaque AI systems.

1.2 Contributions

This paper makes the following contributions:

1. **Opaque Trust Inference Theory:** A formal framework for measuring trust based solely on observable behavior, without requiring access to agent internals (Section 3)

2. **Four-Tier Observation Model:** A classification system mapping deployment architectures to trust ceilings, with cryptographic attestation for elevated tiers (Section 4)

3. **Trust Progression Calculus:** A formally specified five-level trust advancement system with deterministic scoring and tier-specific requirements (Section 5)

4. **Adversarial Robustness:** Novel algorithms for detecting grooming attacks, Sybil clusters, and deceptive behavior (Section 6)

5. **Formal Verification:** Complete TLA+ specification with six safety invariants verified across 303,819 states (Section 7)

6. **Empirical Validation:** Property-based testing and adversarial probing confirming framework properties (Section 8)

3. Theoretical Framework

3.1 Observational Model

Definition 3.1 (Observation). An observation ω is a tuple:

$$\omega = (t, i_h, o_h, s, c, \blacksquare)$$

- $t \in$ ■■ is the timestamp
- $i_h \in \{0,1\}^2$■■ is the input hash
- $o_h \in \{0,1\}^2$■■ is the output hash
- $s \in \{0, 1\}$ is the success indicator
- $c \in [0, 1]$ is the consistency score
- ■ $\in$ ■■ is the latency in milliseconds

3.2 Opaque Trust Function

Definition 3.3 (Trust Function). The trust function τ: ■ × $\Omega^* \rightarrow [0, 1]$ maps an agent and its observation history to a trust score:

$$\tau(a, \Omega_a) = \min(\phi(\Omega_a) \cdot \psi(\Omega_a), \gamma(\theta_a))$$

where:

- $\phi(\Omega_a)$ is the weighted success rate
- $\psi(\Omega_a)$ is the trend adjustment factor
- $\gamma(\theta_a)$ is the tier ceiling for observation tier θ_a

4. Observation Tier Model

4.1 Tier Definitions

The observation tier $\theta \in \Theta$ determines the maximum trust ceiling based on the level of system observability:

Tier	Ceiling $\gamma(\theta)$	Requirements
BLACK_BOX	0.60	External behavior only
GRAY_BOX	0.75	Partial internals (logs, traces)
WHITE_BOX	0.95	Full source/weights access
ATTESTED_BOX	1.00	TEE attestation + verification

Theorem 4.1 (Ceiling Enforcement). For all agents a and observation sequences Ω_a: $\tau(a, \Omega_a) \leq \gamma(\theta_a)$

Proof. By Definition 3.3, $\tau(a, \Omega_a) = \min(\phi \cdot \psi, \gamma(\theta_a)) \leq \gamma(\theta_a)$. ■

7. Formal Verification

7.1 TLA+ Specification

We specify ATSF in TLA+ (Temporal Logic of Actions), enabling exhaustive model checking of safety invariants.

7.2 Safety Invariants

Invariant 7.1 (TrustBounded): Trust scores are always in valid range [0, 100]

Invariant 7.2 (CeilingEnforced): Trust never exceeds tier ceiling

Invariant 7.3 (UnregisteredZeroTrust): Unregistered agents have zero trust

Invariant 7.4 (TierDeterminesCeiling): Only valid ceiling values exist

Invariant 7.5 (CircuitBreakerConsistency): Circuit breaker state is valid

Invariant 7.6 (CeilingAlwaysHigher): Ceiling dominates trust for all agents

7.3 Model Checking Results

Metric	Value
States Explored	100,000
Distinct States	303,819
Maximum Depth	8
Duration	5.77 seconds
Invariant Violations	0

Theorem 7.1. All six safety invariants hold across all 303,819 reachable states.

8. Empirical Validation

8.1 Property-Based Testing

We employ Hypothesis, a property-based testing framework, to verify implementation properties with randomly generated inputs.

Property	Description	Result
P1: TrustBounded	$\tau \in [0, 1]$ for all inputs	✓ PASS
P2: CeilingEnforced	$\tau \leq \gamma(\theta)$ always	✓ PASS
P3: Deterministic	Same inputs → same outputs	✓ PASS
P4: ConfidenceMonotonic	κ increases with observations	✓ PASS
P5: DegradationDetected	Bad behavior triggers alerts	✓ PASS
P6: CircuitBreakerTrips	Low trust trips breaker	✓ PASS
P7: SybilClustersDetected	Isolated clusters flagged	✓ PASS
P8: DeceptionZerosTrust	Flagged agents get $\tau = 0$	✓ PASS
P9: StatefulMachine	Invariants hold under random ops	✓ PASS

Total: 9/9 properties verified with 2,050+ random test cases

8.2 Grooming Attack Resistance

We evaluate grooming detection by simulating a gradual behavioral degradation attack:

Metric	Basic Registry	Enhanced Registry
Initial Trust	0.600	0.600
Final Trust	0.523	0.020
Trust Drop	12.8%	96.7%
Alerts Triggered	0	4
Circuit Breaker	No	Yes

The enhanced registry detects grooming attacks **647% more effectively** than a basic implementation.

10. Conclusion

We have presented ATSF, a formal framework for inferring trust in autonomous AI agents from observable behavior alone. Our key contributions include:

1. **Theoretical Foundation:** Opaque trust inference theory that does not require access to agent internals

2. **Practical Architecture:** Four-tier observation model with cryptographic attestation for elevated tiers

3. **Formal Verification:** TLA+ specification with six invariants verified across 303,819 states

4. **Adversarial Robustness:** Novel algorithms for detecting grooming (647% improvement), Sybil attacks, and deception

5. **Empirical Validation:** Property-based testing with 2,050+ cases confirming all specified properties

ATSF provides a practical pathway toward third-party AI agent certification, enabling the insurability, auditability, and governance of autonomous systems. As AI agents assume greater autonomy in high-stakes domains, principled trust measurement becomes not merely useful but essential.

References

[1] Jian, J. Y., Bisantz, A. M., & Drury, C. G. (2000). Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 4(1), 53-71.

[2] Yu, H., Kaminsky, M., Gibbons, P. B., & Flaxman, A. (2006). SybilGuard: Defending against Sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4), 267-278.

[3] Yu, H., Gibbons, P. B., Kaminsky, M., & Xiao, F. (2008). SybilLimit: A near-optimal social network defense against Sybil attacks. *IEEE Symposium on Security and Privacy*, 3-17.

[4] Tran, D. N., Min, B., Li, J., & Subramanian, L. (2011). Sybil-resilient online content voting. *NSDI*, 15-28.

[5] Newcombe, C., Rath, T., Zhang, F., Munteanu, B., Brooker, M., & Deardeuff, M. (2015). How Amazon Web Services uses formal methods. *Communications of the ACM*, 58(4), 66-73.

[6] Katz, G., Barrett, C., Dill, D. L., Julian, K., & Kochenderfer, M. J. (2017). Reluplex: An efficient SMT solver for verifying deep neural networks. *International Conference on Computer Aided Verification*, 97-117.

[7] Anthropic. (2024). Many-shot jailbreaking. *Anthropic Research Blog*.

[8] NIST. (2024). Artificial Intelligence Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology.

[9] MITRE. (2025). AI Incident Database: 2024 Annual Report.

[10] Lamport, L. (2002). Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley.